



KIM GEBRUIKERS CLUB

NEDERLAND

een club van 6502 gebruikers

Het doel van de vereniging is: het bevorderen van de kennisuitwisseling tussen de gebruikers van 6502-computers, zoals KIM, SYM, JUNIOR, AIM-65, SYSTEM 65, ACORN, PET en CBM, APPLE, CHE-1, ATARI, de Proton-computers, ITT 2020, PEARCOM, PC-100, OSI, BBC, VIC-20 en COMMODORE 64, BASIS 108, PALLAS, SBC 65/68, FORMOSA, ORIC-1, ACE 1000, KEMPAC System 4, NCS 6502, MINTA, enz. enz.

Deze kennisuitwisseling wordt o.a. gerealiseerd door zesmaal per jaar DE 6502 KENNER te publiceren, vijf maal per jaar clubbijeenkomsten te houden, een cassettebibliotheek in stand te houden en papierware-service te bieden. Daarnaast organiseren de leden zelf regionale bijeenkomsten.

Verschijningsdata
DE 6502 KENNER
=====

derde zaterdag
van de maanden:
februari
april, juni
augustus
oktober
december

Regio Den Haag/R'dam e.o.
=====

Inl. Gerard van Roekel
v.d. Palmstraat 11-C
3135 LK Vlaardingen
Tel.: 010 - 351101

Bijeenkomsten
van de club
=====

derde zaterdag
van de maanden:
januari
maart
mei
september
november

Regio Arnhem e.o.
=====

1e en 3e Ma-avond v.d. maand
in HET DORP
(zaal wordt aangegeven in de
hal van de personeelsflat)
Inl.: Henk Deutekom
Tel.: 08303 - 8987

De KIM GEBRUIKERS CLUB NEDERLAND is een volledig onafhankelijke vereniging met statuten en een bestuur. De club is ingeschreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar, onder nummer 634305.

Samenstelling van het bestuur is thans als volgt:

Voorzitter:	<u>Dagelijks bestuur:</u> Rinus Vleesch-Dubois Fl. Nightingalestraat 212 2037 NG Haarlem (023 - 330993)
Erevoorzitter:	Siep de Vries
Sekretaris:	Ruud Uphoff Voorburgpad 10 6843 EM Arnhem (085 - 816935) (19.00 - 20.00)
Penningmeester:	Riet Uphoff - Burghoorn voor adres: zie bij sekretaris
Ledenadministratie:	<u>Leden:</u> Ruud Uphoff (zie bij sekr.)
Redactie DE 6502 KENNER	Willem L. van Pelt Jacob Jordaensstraat 15 2923 CK Krimpen a.d. IJssel (01807 - 19881)
Hard-/Software:	Adri Hankel (05490 - 51151) John van Sorang (01807 - 20589) Erwin Visschedijk (05490 - 71416) Jan Tompot (070 - 931417)
Promotie:	Mw. H. de Vries - Van der Winden
Ereleden:	Anton Mueller zie adres sekretaris
Adres van de vereniging:	Fl.45. = per kalenderjaar, postreke-
Lidmaatschappen:	ning 3757649 t.n.v. KIM Gebruikers Club te Arnhem
Advertentietarieven:	Op aanvraag bij de redactie

DE 6502 KENNER is een uitgave van de KIM Gebruikers Club Nederland.

Adres voor het inzenden van en reacties op artikelen voor DE 6502 KENNER:

Willem L. van Pelt.
Jacob Jordaensstraat 15.
2923 CK KRIMPEN AAN DEN
IJSSSEL.
Tel.: 01807 - 19881

Redactie DE 6502 KENNER:
Vaste medewerkers:

Willem van Pelt
Gerard van Roekel
Frans Smeehuijzen
Jaap van Toledo

Freelance medewerkers:

Frans Bakx
Adri Hankel
Fridus Jonkman
Gert Klein
Herman Kuyvenhoven
Roger Langeveld
Anton Mueller
Koen van Nieuwenhove

Gehele of gedeeltelijke overname van de inhoud van DE 6502 KENNER zonder toestemming van het bestuur is verboden. Toepassing van gepubliceerde programma's, hardware etc. is alleen toegestaan voor persoonlijk gebruik.

DE 6502 KENNER verschijnt 6 x per jaar en heeft een oplage van 500 exemplaren.

Copyright (C) 1984 KIM Gebruikers Club Nederland.

De voorpagina is een aquarel van een KIM, geschilderd door:

Rinus Vleesch Dubois

INHOUDSOPGAVE DE 6502 KENNER NR. 34 OKTOBER 1984

1. Van de redactie	2.
2. UITNODIGING Ledenvergadering/Landelijke Bijeenkomst	22.
3. MOSER Assembler Wordprocessing with Carl Moser's Text Editor ... Hans Buurman	3.
4. C-64 Cassette-interface: simoel ... Fer Weber C.B.M. 64 Kernal Test Routine ... Gerard van Roekel	32. 47.
5. BASIC Tokenized Microsoft Basic Keywords and Addresses SYM-1 ... Willem L. van Pelt Save en Load Routines Proton DOS Voortzetting serie uitbreidingen SYM-Basic oo JUNIOR ... Frans Smeehuijzen Maanlander. Basicoprogramma. ... Maarten van Hintum	34. 36. 42.
6. FORTH Kleintje FORTH. Rekaties. Hex/Ascii-dump ... Gert Klein Korrektheid ... Fridus Jonkman	21. 33. 48.
7. Micro-ADE Patch oo Micro-ADE ... Alfons v.d. Meutter, België	46.
8. COMAL.*KGN Programma Staafdiagram Staafdiagram afdrukken met Microline 80 printer ... Frans Bakx	23.
9. SENIOR-Monitor Patches on Senior Monitor V4.6 Voortzetting van een serie aanpassingen ... Rob Banen Versneld laden Proton Basic programma's ... Gert van Dobroek	26. 48.
10. OS65D Tios & Tricks ... met dank aan Wout van Dinther Wallbreaker for JUNIOR with OHIO-DOS Modifikatie van eerder gepubliceerd progr. Willem Kuitens ... Henk Mocking BUG in DOS-uitbreiding Elektuur, sept. 1984 ... Gert Klein	28. 39. 48.
11. HARDWARE JUNIOR 8K of JUNIOR 64K ... Hans Mooij	44.
12. DIVERSEN Checktaoe ... Phons Bloemen	29.
13. MARKTINFO 65816 microprocessor Elektuurs SAMSOM-65 DOS Computer met OS65D FATE 65	21. 48. 2.
14. VRAAG EN AANBOD	21.24.

Een van de heerlijke dingen die het geweten van een redakteur bij voortduring beroeren is de vraag of hij het zijn lezers/leden van de club wel steeds naar de zin maakt. De realiteit waarschuwt hem daarbij keer op keer voor al te groot optimisme. Immers, zoveel hoofden zoveel zinnen. Of, in het eigen jargon: er is altijd wel een lezer die zich tekort gedaan voelt. Als dat een redakteur te ernstig aan het geweten knaagt, dan schiet hij mijns inziens te kort. Het redaktiewerk kan niet goed functioneren als we alsmaar bezig moeten zijn het iedereen naar de zin te maken, het functioneert pas optimaal als we proberen het zoveel mogelijk lezers naar de zin te maken.

Desalniettemin is het goed dat lezers bij tijd en wijle de redakteur aanspreken op zijn geweten, bijvoorbeeld door te vragen "waarom komt in zo'n oer-nederlands blad de engelse taal voor?". Het is goed om dan eens op een rij te zetten hoe dat in elkaar steekt. Waarom gebruiken we eigenlijk niet uitsluitend het nederlandse?

Nu heb ik nogal de neiging te abstraheren alvorens de kern van de zaak te preciseren. Soms is dat grappig, soms stom- vervelend. Beide risico's neem ik als ik me afvraag wat er nu feitelijk gevraagd wordt. Wat bedoelt men met de nederlandse taal? Gaat het om de taal waarin geen uit andere talen overgenomen ver-nederlandste woorden voorkomen? Hoe ziet eigenlijk het nederlandse van vandaag eruit? Ligt het nederlandse van morgen, door opname van allerlei woorden uit andere talen, niet veel meer voor de hand?

Nog afgezien van het feit dat goede beantwoording van de vraag bemoeilijkt kan worden door onduidelijk gedefinieerde probleemstellingen, het kan ook betekenen dat zo'n eenvoudige vraag voortvloeit uit het gemak dat de mens dient. Er is immers minder inspanning nodig als je uitsluitend de nederlandse taal bezigt.

De redactie heeft te maken met een aantal harde gegevens, welke mijns inziens onomkeerbaar zijn. Ik zal ze hieronder opsommen, maar vraag begrip voor het feit dat slechts de belangrijkste worden genoemd. Deze harde gegevens liggen ingebed in de strijd om de toekomst van blad en club te vergemakkelijken.

- De communicatie in de computerwereld wordt gedomineerd door het engels. Niet enkel vanwege de oorsprong van de activiteiten, maar vooral omdat het engels door vrijwel iedereen wordt begrepen en geaccepteerd.
- In den beginne bestond de club uit oer-nederlandse leden. Het lidmaatschap staat echter open voor iedereen, dus ook van buiten Nederland. We herbergen nu ook leden uit Duitsland, België, Frankrijk, Spanje, Mexico, Portugal, en in India bestaat zelfs belangstelling voor de club. In sommige gevallen soelde Elektuur hier een rol spelen.
- Kennisuitwisseling - doel van onze club - is een mondiale aangelegenheid. Wat onze lezers kunnen leren van de anderen wordt niet bepaald door landsgrenzen.
- Ons blad werd vroeger voornamelijk gevuld door bestuurs leden en redaktied medewerkers. Een enkel lid waande het copy in te sturen. Dat is niet meer. De leden/lezers hebben begrepen dat zij het blad moeten vullen en dat de redactie sturend en stimulerend begeleiding geeft. Onder deze inzenders bevinden er zich die geen woord nederlands kennen, of die geheel of gedeeltelijk in het engels ommaken.

Aangezien nu de inzenders zelf de inhoud van het blad bepalen ligt hier de beantwoording van de vraag vanwaar het engels in ons blad. Oer-engels of steenkolen-engels.

W.L. van Pelt

Een tijdje geleden is bij Adri Hankel en mij een wild idee geboren van een soort extra lande club bijeenkomst. Deze zou dan een heel week-end kunnen duren. Overdag de computers aan en 's avonds gezellig bij elkaar zitten rond de barbecue. In het vorige nummer is hierover ook al iets vermeld. Bij zendamateurs blijkt zoiets levensvatbaar. Zodra Adri en ik, of de redactie, reacties ontvangen gaan we bezien op welke wijze we tot realisering kunnen komen.

Er wordt door mij tussendoor ook nog gewerkt aan een monitor voor de nieuwe CPU-kaart uit Elektuur. Dit stuk bulksoftware begint al aardige vormen aan te nemen. Als het aan mij ligt kan de monitor algemeen cluboed worden (red: al eerder publiceerde J.J. Janssen uit Nijmegen zijn software voor de VDU-kaart: de redactie speurt naar nog meer leuke dingen voor beide kaarten).

Nog even een waarschuwing. In de ELEKTUUR staat een wijziging op de nieuwe CPU-kaart voor 65C02 bezitters. Met deze wijziging wordt voorkomen dat er streepjes op het beeldscherm komen als de CPU in het beeldschermram schrijft. Dit wordt gedaan met behulp van het lijntje RDY op de processor. Dit lijntje wordt met een stukje logica omlaag getrokken als de processor in het VDU-ram wil schrijven terwijl disolav enable op CRTC hoog is. Deze truc werkt aardig als je geen eeprom programmer van Elektuur gebruikt. De eeprom programmer maakt ook gebruik van de RDY-lijn. En tot mijn grote verbazing zag ik dat zowel de lijn op de VDU-kaart als de lijn op de eeprom programmer gestuurd worden door TTL poorten met een totelpaal uitoang. Dit is vragen om moeilijkheden. De oplossing ligt voor de hand. Zorg ervoor dat de TTL-poorten die de RDY-lijn moeten sturen een open collector uitoango hebben. Er is op beide kaarten een open collector-poort meer vrij, dus moet er eens een 74LS05 of iets dergelijks bij op gemonteerd worden. Veel succes ermee.

Erwin Visschedijk.

12 K FATE 65 FOR JUNIOR.

Format lister
Assembler
Tool-utilities
Editor

FATE 65 has the editor as a central program. With the editor one is able to develop text-files and source-files, to write on tape and to read from tape. With the format lister text-files are made on a printer. The lister recognises several commands, under which control-commands for the OKI Microline 80 printer. With the assembler source-files you can convert to object-code. From cassette to cassette. From memory to memory. From cassette to memory. From memory to cassette. Merging is possible. Splitting is possible. You need an extended JUNIOR-computer, two cassette recorders with motor-control, and at least 16 K RAM, better 32 K (2000-9000). Ask Mr. W.L. van Pelt, c/Jacob Jordaensstraat 15 Krimpen a.d. IJssel, 2923 CK, The Netherlands, for a list of prices.

WORD PROCESSING WITH CARL MOSEY'S TEXT EDITOR

The 6502 macro assembler has a text editor that is well suited for its purpose, setting up and maintaining its sourcefile. However, to edit other sorts of text a different program is needed. The program described here enables the user to insert commands in the text which allow him to have the text outputted in a desired form, both on screen as on a printer. Here, the oki microline 80 is used, but the program can easily be modified to fit to other printers.

1. General description of the program.

First, some of the assemblers vectors must be initialised by using RUN address, where the value of address depends on the programs assembly. During use, the text can be outputted with the USER command. If the break key is used, outputting can be resumed with CTRL Y. If this key is pressed at any other moment, the program will resume where it stopped last time, or after initialisation at the start of the textfile. However in the last case no output will be seen because both output devices have been switched off. After an error has been detected, CTRL Y will cause a restart at the start of the line in which the error occurred.

Commands must be enclosed between square brackets, two symbols with ascii values of hex 5B and 5D. These characters are printed by the microline as + and -. After a + is found, the next character is supposed to be the command. Commands must be followed by a colon or closing bracket, the former meaning another command follows, the latter ending the commands. The twenty-five recognised commands are explained below. It must be stressed that some of the commands are only effective at the end of a line. If commands set or reset a printing mode, the last time this command was used, holds. E.g. in a line the last H command encounter will determine the printer mode.

2. Printing procedure.

At the end of a line, or on moving to another line, the current line is printed. This may take three passes:

1-On the first pass, the line is printed as usual.

2-On the second pass, if the 2 or B command has been used, desired parts of the line are printed again in the desired form. This second pass allows words to be printed extra dark or in a special way (e.g. the dutch word reëel).

3-On the third pass, if a U command has been used, an underline will be added. On default this will use the next line completely as is shown in this text's headings, but other settings can be obtained by using the C command.

The last two phases will only be used when needed, but they will all be used when a graphic code ()127) has been used. This is because the microline has an advised duty cycle of 40% when printing graphics.

3. Spaces.

During output spaces are subject to the following rules:

1-Between two groups of non-space characters only one space is allowed. More spaces can however be obtained with several commands, such as S, T and A.

2-At the end of each line in the original text, a space is inserted. However, rule 1 must be obeyed. Paragraphs are available with the P command. The first line of a paragraph is tabulated. The number of spaces here can be changed with the J command. It must be remarked that all above commands are useless when the lines are filled as in the next paragraph, after the Q command is given.

4. Line sizes.

The length of a line is determined by the current printer mode and the positions of the borders. The printer mode is selected with de H and V commands for width and letter size, and letter height respectively. Columns are numbered from left to right with the leftmost being nr. 0, and the L and R commands set the borders to their values. The borders are reset after a H command. The T and . commands tabulate to a certain column. A terminal width can be given, if the width of a line is equal to this, the linefeed on screen will be suppressed to improve layout.

5. Pages

After an F command, printing is done in pages. A page number is printed at the top of each page. The # command sets the page number to a value, which of course is a decimal. The I command allows the user to determine the size of the pages himself. It is assumed that V0 gives 6 lines/inch and V1 gives 8 lines/inch, so that the space between pages can be given in either 1/6 or 1/8 inch, the latter by simply adding 128. A value of 255 gives only a halt at the end of each page. The size of the page must be given in inches as an integer number. No assumptions made here. Only if an F command is given very close to the end of a page slight errors of a fraction of an inch will be made.

5. Miscellaneous.

The O command allows output to be sent to printer, screen neither or both, and the K command just waits for a key to be pressed.

Of course, sometimes an illegal input will be found. Most possibilities (as far as the author could see!) are quite harmless and will be ignored. If possible, unnecessary bits are thrown away and tabulations to not existing columns aren't executed, but some things just cannot be allowed. In that case, an error message will be issued by means of the standard routine, showing !xx AT LINE yyyy/zz. In extreme cases yyyy may point to the line after the one containing the error. The zz field shows the value that caused the error. All error codes are shown in a table.

Detailed description of all commands.

The following notations will be used:

n or m.....Either a decimal integer or the symbol ' followed by any character (this gives the ascii-value).

strings.....A sequence of characters not containing : or <, ended by : or <.

1-Commands without operands.

- D Move to the next line.
- P Start a new paragraph: move to next line and tabulate.
- F If pages weren't used: start using pages, begin a new page (turned off by I).
- K If pages were used: start a new page.
- Wait for any key depressed.
- 2 Start printing everything twice (extra dark).
- 1 Print everything once.

Defaults are: no pages, print once.

2-Commands with one operand.

- Ln Set left hand border to n.
- Rn Set right hand border to n.
- Hn Set horizontal mode. Modes 0-2 give compressed, normal and double width on a long line, modes 3-5 the same on a short line.
- Vn Set vertical mode. Mode 0 gives 6 lpi, mode 1 gives 8 lpi. Dat for the H and V commands are contained in a table at the end of the program.
- Wn Set terminal width.
- Tn Tabulate to column n. If the cursor is past that column, the command is ignored.
- .n Same as the T command, only .s are printed.
- On Output control. n=0 means no output, n=1 uses the terminal, n=2 uses the printer and n=3 uses both.
- Jn Set the paragraph tabulation to n.
- Sn Print n spaces.
- En Error control. n=0 means continue after error, n(>0) causes a halt. See the assemblers .CE pseudo op.
- In Set the page length to n inch. There is also a two operand version of this command. I0 or I turns the pages off.
- +n Add n to all following ascii codes. This allows use of graphic codes
- #n Set the page number to n.
- An Send ascii code n.
- Un Underline with symbol n. Useful are =, _ , - and ~. Switched off with U.
- Qn Fill out all lines (see fourth and fifth paragraph) unless the line has less than n words.
- Bstring Print string, starting at current position, on the second pass.

Defaults are: H0;V1;I11;255;Q:#0;E1;01;J5;+0;U.

3-Commands with two operands:

- In,m Set page length to n inch, independent of printer mode. Set space between pages to m times 1/6 inch if m<128.
- Set space between pages to (m-128) times 1/8 inch, if 127<m<255. If m=255, just stop printing after each page.
- Cn,m If m(>0), send a carriage return after printing pass n, if m=0 don't.

Error codes.

Following error codes are used:

- BC Bad Command. Shown is the ascii value in hex.
- 30 Border setting out of range. Shown is the value of the L or R command involved.
- 31 Paragraph tabulation out of range. May occur after a L, R, or J command. Shown is the value of the last J command.
- 32 Tabulation out of range. A T or . command has been given to a position on the right of the right hand border. Shown is the operand.
- 33 Syntax error in a C command, or the first operand is zero. Shown is 00 in the last case, or an ascii value in the first.

All values are shown in hex.

PA

```

0001 :PRINTER FACILITIES PROGRAM
0002 .BA $9000
0003 BUFLN .DE 132 :LARGEST NUMBER OF CHARACTERS/LINE POSSIBLE
0004 OUTCH .DE $C80 :CHARACTER TO SCREEN WITHOUT BREAKTEST
0005 RECHA .DE $12AE :READ CHARACTER FROM KEYBOARD
0006 PRINTER .DE $400C :CHARACTER TO PRINTER ONLY
0007 BREAK .DE $1A80 :BREAKTEST ADDRESS: BIT 7 IS 0 FOR BREAK
0008 WARMST .DE $2003 :ASSEMBLER WARM START
0009 ERRORM .DE $244B :ISSUE ERROR MESSAGE (ASSEMBLER)
0010 :ERROR FORM IS !XX AT LINE YYYY/ZZ WHERE XX IS X-REG.
0011 LINENR .DE $108 :VARIABLE FOR YYYY
0012 FILENR .DE $128 :SAME FOR ZZ
0013 CONTFI .DE $112 :FLAG ( ) 0 AFTER .CE
0014 END .DE $D3 :END OF TEXTFILE ADDRESS
0015 BEGIN .DE $100 :BEGIN OF TEXTFILE
0016 PNT .DE $20 :POINTER TO TEXTFILE
0017 :INIT - CHANGE ZERO PAGE VECTORS
0018 :CTRL Y GIVES JUMP TO $0
0019 :USER GIVES JUMP TO $3
9000- A2 05 0020 INIT LDX #5
9002- BD 18 90 0021 INITO LDA TABEL1.X
9005- 95 00 0022 STA *0,X
9007- CA 0023 DEX
9008- 10 F8 0024 BPL INITO
0025 :SWITCH OFF OUTPUTS
900A- A9 00 0026 LDA #0
900C- 8D 1E 9A 0027 STA SCREEN
900F- 8D 1D 9A 0028 STA PRINT
0029 :INITIALISE VARIABLES
9012- 20 A5 90 0030 JSR INIT2
9015- 4C 03 20 0031 JMP WARMST
0032 :TABEL1 - ZERO PAGE DATA
9018- 4C 0033 TABEL1 .BY $4C
9019- 1E 90 0034 .SI CTRL Y
901B- 4C 0035 .BY $4C
901C- 2B 90 0036 .SI USER
0037 :CTRL Y - SEND CTRLF AND CONTINUE
901E- A9 0D 0038 CTRL Y LDA #$D
9020- 20 3C 90 0039 JSR OUT
9023- A9 0A 0040 LDA #$A
9025- 20 3C 90 0041 JSR OUT
9028- 4C 04 92 0042 JMP LOOP
0043 :USER - ENABLE SCREEN. DISABLE PRINTER AND START
902B- A9 00 0044 USER LDA #0
902D- 8D 1D 9A 0045 STA PRINT
9030- 49 01 0046 EOR #$1
9032- 8D 1E 9A 0047 STA SCREEN
9035- 4C 01 92 0048 JMP MAIN
0049 :OUTO - SEND CHARACTER TO SCREEN WITHOUT LF-SUPPRESSION
9038- AA 0050 OUTO TAX
9039- 4C 51 90 0051 JMP OUT1
0052 :OUT - SEND CHARACTER TO SCREEN
903C- AA 0053 OUT TAX
903D- 10 06 0054 BPL SUPTEST
0055 :ASCII > 127 : GRAPHIC CODE. REDUCE DUTY CYCLE
903F- 8D 11 9A 0056 STA USED2
9042- 8D 13 9A 0057 STA UNDERLD
9045- C9 0A 0058 SUPTEST CMP #$A
9047- D0 08 0059 BNE OUT1
0060 :LINEFEED : IF WIDTH=TERMINAL WIDTH, DON'T SEND TO TERMINAL
9049- AD 14 9A 0061 LDA HSIZE
904C- CD 22 9A 0062 CMP TWIDTH
904F- F0 09 0063 BEQ OUT2
0064 :SEND TO TERMINAL ?
9051- AD 1E 9A 0065 OUT1 LDA SCREEN
9054- F0 04 0066 BEQ OUT2
9056- 8A 0067 TXA
9057- 20 80 0C 0068 JSR OUTCH
0069 :SEND TO PRINTER ?
905A- AD 1D 9A 0070 OUT2 LDA PRINT
905D- F0 04 0071 BEQ ENDOUT
905F- 8A 0072 TXA
9060- 20 0C 40 0073 JSR PRINTER
9063- E0 0A 0074 ENDOUT CPX #$A

```


9065-	D0 03	0075	BNE ENDOUT2
		0076	:LINEFEED : CHECK IF END OF PAGE
9067-	20 71 95	0077	JSR NEWLINE
906A-	60	0078	RTS
		0079	:NEXT - GET NEXT CHARACTER FROM 'MOSER' FILE
		0080	:CHECK IF END OF TEXT
906B-	A5 21	0081	NEXT LDA *PNT+1
906D-	C5 D4	0082	CMP *END+1
906F-	D0 04	0083	BNE NEXT99
9071-	A5 20	0084	LDA *PNT
9073-	C5 D3	0085	CMP *END
		0086	:END OF TEXT : CARRY SET AND EXIT
9075-	B0 2B	0087	NEXT99 BCS NEXTEND
		0088	:GET NEXT CHARACTER, SAVE COPY IN CCHAR
9077-	A0 00	0089	LDY #0
9079-	A2 01	0090	LDX #1
907B-	B1 20	0091	LDA (PNT),Y
907D-	8D 2C 9A	0092	STA CCHAR
9080-	10 0E	0093	BPL NEXT1
		0094	:BIT 7 SET MEANS END OF LINE
		0095	:GET LINE NUMBER
9082-	C8	0096	INY
9083-	B1 20	0097	LDA (PNT),Y
9085-	8D 08 01	0098	STA LINENR
9088-	C8	0099	INY
9089-	B1 20	0100	LDA (PNT),Y
908B-	8D 09 01	0101	STA LINENR+1
908E-	A2 03	0102	LDX #3
		0103	:ADVANCE PNT TO NEXT CHARACTER IN TEXT
9090-	18	0104	NEXT1 CLC
9091-	8A	0105	TXA
9092-	65 20	0106	ADC *PNT
9094-	85 20	0107	STA *PNT
9096-	A5 21	0108	LDA *PNT+1
9098-	69 00	0109	ADC #0
909A-	85 21	0110	STA *PNT+1
		0111	:FETCH COPY AND MASK BIT 7
909C-	AD 2C 9A	0112	LDA CCHAR
909F-	29 7F	0113	AND #01111111
90A1-	60	0114	RTS
		0115	:END OF TEXT : RETURN TO 'MOSER'
90A2-	20 85 91	0116	NEXTEND JSR SENDBUFS
		0117	:INIT2 - INITIALISE ALL VARIABLES AND PRINTER
90A5-	AD 00 01	0118	INIT2 LDA BEGIN
90A8-	AC 01 01	0119	LDY BEGIN+1
		0120	:PNT TO START OF TEXT
90AB-	85 20	0121	STA *PNT
90AD-	84 21	0122	STY *PNT+1
		0123	:AND LINENR TO FIRST LINE
90AF-	A0 00	0124	LDY #0
90B1-	B1 20	0125	LDA (PNT),Y
90B3-	8D 08 01	0126	STA LINENR
90B6-	20 8F 92	0127	JSR INCPNT
90B9-	B1 20	0128	LDA (PNT),Y
90BB-	8D 09 01	0129	STA LINENR+1
		0130	:FIRST CHARACTER IS AFTER LINE NUMBER
90BE-	20 8F 92	0131	JSR INCPNT
		0132	:PARAGRAPH TABULATION = 5
90C1-	A9 05	0133	LDA #5
90C3-	8D 21 9A	0134	STA PTAB
		0135	:TERMINAL WIDTH = 64
90C6-	A9 40	0136	LDA #64
90C8-	8D 22 9A	0137	STA TWIDTH
		0138	:DON'T USE SECOND BUFFER, UNDERLINE, PAGES
90CB-	A9 00	0139	LDA #0
90CD-	8D 10 9A	0140	STA USE2
90D0-	8D 12 9A	0141	STA UNDERL
90D3-	8D 1B 9A	0142	STA PAGED
		0143	:DON'T FILL OUT, START AT TOP OF PAGE 0
90D6-	8D 2A 9A	0144	STA FILL
90D9-	8D 19 9A	0145	STA VCNT
90DC-	8D 1A 9A	0146	STA VCNT+1
90DF-	8D 1C 9A	0147	STA PAGE
		0148	:NORMAL CHARACTERS, NO LINEFEED AFTER PASS 1, COMPRESSED MODE
90E2-	8D 23 9A	0149	STA OFFSET
90E5-	8D 27 9A	0150	STA CTRL
90E8-	20 11 91	0151	JSR HMODE

90EB-	A9 01	0152		:DON'T CONTINUE AFTER ERRORS. 8 LINES/INCH
90ED-	8D 12 01	0153		LDA #1
90F0-	20 3B 91	0154		STA CONTFI
		0155		JSR VMODE
		0156		:LINEFEED AFTER PASS 2 AND 3
90F3-	A9 0A	0157		LDA #A
90F5-	8D 28 9A	0158		STA CTRL+1
90F8-	8D 29 9A	0159		STA CTRL+2
		0160		:PAGES OF 11 INCH
90FB-	A9 02	0161		LDA #2
90FD-	8D 25 9A	0162		STA PSIZE+1
9100-	A9 10	0163		LDA #10
9102-	8D 24 9A	0164		STA PSIZE
		0165		:STOP AFTER EACH PAGE
9105-	A9 FF	0166		LDA #255
9107-	8D 26 9A	0167		STA BSIZE
		0168		:CLEAR BUFFERS
910A-	20 5D 91	0169		JSR CLEARBUFS
		0170		:SEND CR AND LF
910D-	20 85 91	0171		JSR SENDBUFS
9110-	60	0172		RTS
		0173		:HMODE - SET CHARACTER AND LINE WIDTH
9111-	8D 2E 9A	0174	HMODE	STA TEMP
9114-	8D 16 9A	0175		STA HM
		0176		:MULTIPLY ACCU BY 5
9117-	0A	0177		ASL A
9118-	0A	0178		ASL A
9119-	18	0179		CLC
911A-	6D 2E 9A	0180		ADC TEMP
		0181		:SEND FOUR CODES FROM TABLE TO PRINTER
911D-	AA	0182		TAX
911E-	A0 04	0183		LDY #4
9120-	BD 33 9A	0184	HM1	LDA HOR,X
		0185		:DON'T SEND ZEROES
9123-	F0 03	0186		BEQ HM2
9125-	20 0C 40	0187		JSR PRINTER
9128-	E8	0188	HM2	INX
9129-	88	0189		DEY
912A-	D0 F4	0190		BNE HM1
		0191		:5TH IN TABLE IS LINE WIDTH
912C-	BC 33 9A	0192		LDY HOR,X
912F-	8C 14 9A	0193		STY HSIZE
		0194		:CLEAR BORDERS
9132-	8C 20 9A	0195		STY RBORD
9135-	A9 00	0196		LDA #0
9137-	8D 1F 9A	0197		STA LBORD
913A-	60	0198		RTS
		0199		:VMODE - SET LINES/INCH
		0200		:THE FORM FEED ROUTINE EXPECTS VMODE 1 TO BE 8 LPI
		0201		:AND VMODE 0 NOT
913B-	8D 2E 9A	0202	VMODE	STA TEMP
913E-	8D 17 9A	0203		STA VM
		0204		:MULTIPLY ACCU BY 5
9141-	0A	0205		ASL A
9142-	0A	0206		ASL A
9143-	18	0207		CLC
9144-	6D 2E 9A	0208		ADC TEMP
		0209		:SEND FOUR CODES (BUT NO ZEROES) TO PRINTER
9147-	AA	0210		TAX
9148-	A0 04	0211		LDY #4
914A-	BD 5B 9A	0212	VM1	LDA VER,X
914D-	F0 03	0213		BEQ VM2
914F-	20 0C 40	0214		JSR PRINTER
9152-	E8	0215	VM2	INX
9153-	88	0216		DEY
9154-	D0 F4	0217		BNE VM1
		0218		:5TH IN TABLE IS LINE HEIGHT (TIMES 1/48 INCH)
9156-	BD 5B 9A	0219		LDA VER,X
9159-	8D 15 9A	0220		STA VSIZE
915C-	60	0221		RTS
		0222		:CLEARBUFS - CLEAR ALL BUFFERS AND 'USED' FLAGS
915D-	A2 00	0223	CLEARBUFS	LDX #0
915F-	A9 20	0224		LDA #'
9161-	8D 2B 9A	0225		STA LCHAR
9164-	9D 84 98	0226	CLE1	STA BUF1,X
9167-	9D 08 99	0227		STA BUF2,X
916A-	9D 8C 99	0228		STA BUF3,X

916D-	E8	0229	INX
916E-	E0 84	0230	CPX #BUFLEN
9170-	DO F2	0231	BNE CLE1
		0232	:CLEAR THE FLAGS
9172-	AE 10 9A	0233	LDX USE2
9175-	8E 11 9A	0234	STX USED2
9178-	AE 12 9A	0235	LDX UNDERL
917B-	8E 13 9A	0236	STX UNDERLD
		0237	:START AT LEFT BORDER
917E-	AE 1F 9A	0238	LDX LBORD
9181-	8E 18 9A	0239	STX HCNT
9184-	60	0240	RTS
		0241	:SENDBUFS -SEND THE BUFFERS TO SCREEN AND/OR PRINTER
9185-	AD 2A 9A	0242	SENDBUFS LDA FILL
9188-	FO 06	0243	BEQ SE0
		0244	:FILL()0 MEANS FILL OUT LINES
		0245	:FIRST ALL TO THE LEFT
918A-	20 26 98	0246	JSR SWEEP
		0247	:THEN FILL OUT
918D-	20 4D 97	0248	JSR FILLOUT
		0249	:ANYTHING IN BUF1 ?
		0250	:IF NOT, DON'T DO PASS 1
9190-	AE 18 9A	0251	LDX HCNT
9193-	AO 00	0252	LDY #0
9195-	EC 1F 9A	0253	CPX LBORD
9198-	FO 0C	0254	BEQ SE10
		0255	:PASS 1 : PRINT BUF1
919A-	B9 84 98	0256	LDX BUF1.Y
919D-	20 3C 90	0257	JSR OUT
91A0-	C8	0258	INX
91A1-	CC 14 9A	0259	CPY HSIZE
91A4-	DO F4	0260	BNE SE1
		0261	:SEND CR AND IF WANTED LINE FEED
91A6-	A9 0D	0262	LDX #0
91A8-	20 3C 90	0263	JSR OUT
91AB-	AD 27 9A	0264	LDX CTRL
91AE-	FO 03	0265	BEQ SE11
91B0-	20 3C 90	0266	JSR OUT
		0267	:SECOND BUFFER USED ?
		0268	:IF NOT DON'T DO PASS 2
91B3-	AD 11 9A	0269	LDX USED2
91B6-	FO 0E	0270	BEQ SE3
91B8-	AO 00	0271	LDY #0
		0272	:PASS 2 : EXTRA DARK AND SPECIAL USE (e" ETC.)
91BA-	B9 08 99	0273	LDX BUF2.Y
91BD-	20 3C 90	0274	JSR OUT
91C0-	C8	0275	INX
91C1-	CC 14 9A	0276	CPY HSIZE
91C4-	DO F4	0277	BNE SE2
		0278	:SEND CR AND IF WANTED, LF
91C6-	A9 0D	0279	LDX #0
91C8-	20 3C 90	0280	JSR OUT
91CB-	AD 28 9A	0281	LDX CTRL+1
91CE-	FO 03	0282	BEQ SE30
91D0-	20 3C 90	0283	JSR OUT
		0284	:BUF3 USED ? IF NOT DON'T DO PASS 3
91D3-	AD 13 9A	0285	LDX UNDERLD
91D6-	FO 1B	0286	BEQ SE5
91D8-	AO 00	0287	LDY #0
		0288	:PASS 3 : UNDERLINE
91DA-	B9 8C 99	0289	LDX BUF3.Y
91DD-	20 3C 90	0290	JSR OUT
91E0-	C8	0291	INX
91E1-	CC 14 9A	0292	CPY HSIZE
91E4-	DO F4	0293	BNE SE4
		0294	:SEND CR AND IF WANTED, LF
91E6-	A9 0D	0295	LDX #0
91E8-	20 3C 90	0296	JSR OUT
91EB-	AD 29 9A	0297	LDX CTRL+2
91EE-	FO 03	0298	BEQ SE5
91F0-	20 3C 90	0299	JSR OUT
		0300	:BREAKTEST AFTER A LINE HAS BEEN PRINTED
91F3-	2C 80 1A	0301	SE5 BIT BREAK
91F6-	30 06	0302	BMI GOCL
		0303	:BREAK : CLEAR BUFFERS AND GOTO 'MOSER'
91F8-	20 5D 91	0304	JSR CLEARBUFS
91FB-	4C 03 20	0305	JMP WARMST

91FE- 4C 5D 91	0306		:NO BREAK : CLEAR BUFFERS
	0307	GOCL	JMP CLEARBUFS
9201- 20 A5 90	0308	:MAIN -	MAIN PROGRAM
	0309	MAIN	JSR INIT2
9204- AD 2B 9A	0310	:LOOP -	MAIN LOOP : GET CHARACTER, SEND CHARACTER TO BUFFER
	0311	LOOP	LDA LCHAR
	0312		:END OF 'MOSEY' LINE ? THEN EXTRA SPACE !
9207- 10 07	0313		BPL NOXTRASP
9209- A9 20	0314		LDA #'
920B- 8D 2C 9A	0315		STA CCHAR
920E- D0 03	0316		BNE XTRASP
	0317		:ELSE JUST GET NEXT CHARACTER
9210- 20 6B 90	0318	NOXTRASP	JSR NEXT
9213- C9 5B	0319	XTRASP	CMP #'
	0320		:DOES A COMMAND BEGIN HERE ?
9215- D0 03	0321		BNE NOCOM
9217- 4C 17 93	0322		JMP COMM
	0323		:NO COMMAND, SO JUST PUT TO BUFFER
921A- 18	0324	NOCOM	CLC
	0325		:OFFSET IS ADDED FOR GRAPHICS
921B- 6D 23 9A	0326		ADC OFFSET
921E- AA	0327		TAX
	0328		:WAS LAST CHARACTER A SPACE ?
921F- AD 2B 9A	0329		LDA LCHAR
9222- 29 7F	0330		AND #01111111
9224- C9 20	0331		CMP #'
9226- 8D 2B 9A	0332		STA LCHAR
9229- D0 05	0333		BNE NOSP
922B- 8A	0334		TXA
	0335		:AND IS THIS CHARACTER A SPACE ?
922C- C9 20	0336		CMP #'
	0337		:IF BOTH WERE SPACES, FORGET THIS SPACE
922E- F0 D4	0338		BEQ LOOP
9230- 8A	0339	NOSP	TXA
9231- 20 37 92	0340		JSR PUT
9234- 4C 04 92	0341		JMP LOOP
	0342	:PUT -	PUT A CHARACTER TO BUFFER
9237- AE 18 9A	0343	PUT	LDX HCNT
923A- 9D 84 98	0344		STA BUF1,X
	0345		:IF WANTED, TO SECOND BUFFER
923D- AC 10 9A	0346		LDY USE2
9240- F0 03	0347		BEQ NOT2
9242- 9D 08 99	0348		STA BUF2,X
9245- 48	0349	NOT2	PHA
	0350		:REMEMBER THIS CHARACTER
9246- AD 2C 9A	0351		LDA CCHAR
9249- 8D 2B 9A	0352		STA LCHAR
924C- AD 12 9A	0353		LDA UNDERL
924F- F0 03	0354		BEQ NOUNDE
	0355		:IF WANTED, UNDERLINE
9251- 9D 8C 99	0356		STA BUF3,X
9254- 68	0357	NOUNDE	PLA
9255- EE 18 9A	0358		INC HCNT
9258- AC 18 9A	0359		LDY HCNT
925B- CC 20 9A	0360		CPY RBORD
	0361		:RIGHT HAND BORDER REACHED ?
925E- D0 06	0362		BNE ENDPUT
	0363		:IF SO, PRINT LINE
9260- 20 70 92	0364		:DON'T CUT WORDS IN TWO
	0365		JSR CUTOFF
	0366		:PRINT BUFFERS
9263- 20 85 91	0367		JSR SENDBUFS
9266- 60	0368	ENDPUT	RTS
	0369	:DECPNT -	PNT := PNT-1 (TWO BYTE DECREMENT)
9267- A5 20	0370	DECPNT	LDA *PNT
9269- D0 02	0371		BNE DEC1
926B- C6 21	0372		DEC *PNT+1
926D- C6 20	0373	DEC1	DEC *PNT
926F- 60	0374		RTS
	0375	:CUTOFF -	MAKE SURE WORDS AREN'T SPLIT
9270- A0 00	0376	CUTOFF	LDY #0
9272- B1 20	0377		LDA (PNT),Y
9274- 29 7F	0378		AND #\$7F
9276- C9 20	0379		CMP #'
	0380		:IF NEXT CHARACTER IS SPACE : NO PROBLEM

9278-	DO 30	0381	:ELSE GOTO NOSPACE
927A-	60	0382	BNE NOSPACE
		0383	RTS
927B-	20 67 92	0384	:LAST - OPPOSITE OF NEXT : PNT MOVES BACK TO LAST CHARACTER
927E-	20 67 92	0385	LAST JSR DECPNT
9281-	20 67 92	0386	JSR DECPNT
9284-	AO 00	0387	JSR DECPNT
9286-	B1 20	0388	LDY #0
9288-	C9 9F	0389	LDA (PNT),Y
		0390	CMP #\$9F
928A-	BO 0E	0391	:BYTES >\$9F ARE MOSERS END OF LINE BYTES
		0392	BCS LA2
928C-	20 8F 92	0393	:NO END OF LINE, MOVE ONLY ONE BYTE BACK
		0394	JSR INCPNT
928F-	E6 20	0395	:INCPNT - PNT := PNT+1 (TWO-BYTE INCREMENT)
9291-	DO 02	0396	INCPNT INC *PNT
9293-	E6 21	0397	BNE LA1
9295-	B1 20	0398	INC *PNT+1
		0399	LA1 LDA (PNT),Y
		0400	.LI WORDPR.MO2
F02 1C88 4292-5F1A			
		0400	:ANOTHER END OF LINE CHECK
9297-	C9 9F	0401	CMP #\$9F
9299-	60	0402	RTS
		0403	:END OF LINE FOUND: SAVE LINE NUMBER
929A-	C8	0404	LA2 INY
929B-	48	0405	PHA
929C-	B1 20	0406	LDA (PNT),Y
929E-	BD 08 01	0407	STA LINENR
92A1-	C8	0408	INY
92A2-	B1 20	0409	LDA (PNT),Y
92A4-	BD 09 01	0410	STA LINENR+1
92A7-	68	0411	PLA
92A8-	38	0412	SEC
92A9-	60	0413	RTS
		0414	:NO SPACE WAS FOUND: THIS WORD GOES TO THE NEXT LINE
92AA-	20 7B 92	0415	NOSPACE JSR LAST
		0416	:MOVE BACK UNTIL A SPACE OR END OF LINE IS FOUND
92AD-	BO 06	0417	BCS NO1
92AF-	29 7F	0418	AND #\$7F
92B1-	C9 20	0419	CMP #'
92B3-	DO F5	0420	BNE NOSPACE
92B5-	20 6B 90	0421	NO1 JSR NEXT
		0422	:DO THE SAME IN BUF1
92B8-	AE 20 9A	0423	LDX RBORD
92BB-	CA	0424	DEX
92BC-	BD 84 98	0425	NO2 LDA BUF1,X
92BF-	C9 20	0426	CMP #'
92C1-	F0 03	0427	BEQ NO3
92C3-	CA	0428	DEX
92C4-	DO F6	0429	BNE NO2
		0430	:MAKE THE END OF THE LINE ALL SPACES
92C6-	A9 20	0431	NO3 LDA #'
92C8-	9D 84 98	0432	STA BUF1,X
92CB-	9D 08 99	0433	STA BUF2,X
92CE-	9D 8C 99	0434	STA BUF3,X
92D1-	E8	0435	INX
92D2-	EC 20 9A	0436	CPX RBORD
92D5-	DO EF	0437	BNE NO3
92D7-	60	0438	RTS
		0439	:DECIM - READ IN A DECIMAL NUMBER OR ASCII VALUE
92D8-	A9 00	0440	DECIM LDA #0
92DA-	BD 2D 9A	0441	STA NUMB
92DD-	20 6B 90	0442	DECIM1 JSR NEXT
		0443	: FOUND MEANS ASCII CODE FOLLOWS
92E0-	C9 27	0444	CMP #'
92E2-	F0 0C	0445	BEQ ASCII
		0446	:TEST WHETHER IT'S A NUMBER
92E4-	20 B8 96	0447	JSR NUMTEST
92E7-	10 0E	0448	BPL DECIM3
		0449	:NOT A NUMBER, MOVE BACK AND EXIT
92E9-	20 7B 92	0450	JSR LAST
92EC-	AD 2D 9A	0451	LDA NUMB
92EF-	60	0452	RTS
		0453	:ASCII CODE, READ IT AND EXIT

92F0-	20	6B	90	0454	ASCII	JSR NEXT	
92F3-	8D	2D	9A	0455		STA NUMB	
92F6-	60			0456		RTS	
				0457		:IT'S A NUMBER: NUMB=NUMB*10+(THIS NUMBER)	
92F7-	38			0458	DECIM3	SEC	
92F8-	E9	30		0459		SBC ##30	
92FA-	48			0460		PHA	
				0461		:NUMB=10*NUMB MEANS NUMB=2*(NUMB+4*NUMB)	
92FB-	AD	2D	9A	0462		LDA NUMB	
				0463		:MULTIPLY BY FOUR	
92FE-	0E	2D	9A	0464		ASL NUMB	
9301-	0E	2D	9A	0465		ASL NUMB	
9304-	18			0466		CLC	
				0467		:ADD ONE TIME	
9305-	6D	2D	9A	0468		ADC NUMB	
				0469		:MULTIPLY BY TWO	
9308-	0A			0470		ASL A	
9309-	8D	2D	9A	0471		STA NUMB	
930C-	68			0472		PLA	
				0473		:ADD (THIS NUMBER)	
930D-	18			0474		CLC	
930E-	6D	2D	9A	0475		ADC NUMB	
9311-	8D	2D	9A	0476		STA NUMB	
9314-	4C	DD	92	0477		JMP DECIM1	
				0478	:COMM - MAIN LABEL OF COMMAND MODE		
				0479	:WHAT FOLLOWS IS A SKIP LIST OF COMMANDS		
				0480	:IT'S RATHER LONG AND SIMPLE- PLEASE DON'T GET BORED		
9317-	20	6B	90	0481	COMM	JSR NEXT	
931A-	C9	4C		0482		CMP #'L	
931C-	DO	03		0483		BNE NOL	
931E-	4C	E6	93	0484		JMP LCOM	
9321-	C9	52		0485	NOL	CMP #'R	
9323-	DO	03		0486		BNE NOR	
9325-	4C	0A	94	0487		JMP RCOM	
9328-	C9	44		0488	NOR	CMP #'D	
932A-	DO	03		0489		BNE NOD	
932C-	4C	27	94	0490		JMP DCOM	
932F-	C9	48		0491	NOD	CMP #'H	
9331-	DO	03		0492		BNE NOH	
9333-	4C	2D	94	0493		JMP HCOM	
9336-	C9	56		0494	NOH	CMP #'V	
9338-	DO	03		0495		BNE NOV	
933A-	4C	38	94	0496		JMP VCOM	
933D-	C9	55		0497	NOV	CMP #'U	
933F-	DO	03		0498		BNE NOU	
9341-	4C	43	94	0499		JMP UCOM	
9344-	C9	3A		0500	NOU	CMP #'1	
9346-	FO	CF		0501		BEQ COMM	
9348-	C9	32		0502		CMP #'2	
934A-	DO	03		0503		BNE NOT2C	
934C-	4C	68	94	0504		JMP C2COM	
934F-	C9	31		0505	NOT2C	CMP #'1	
9351-	DO	03		0506		BNE NOT1	
9353-	4C	71	94	0507		JMP C1COM	
9356-	C9	57		0508	NOT1	CMP #'W	
9358-	DO	03		0509		BNE NOW	
935A-	4C	79	94	0510		JMP WCOM	
935D-	C9	4A		0511	NOW	CMP #'J	
935F-	DO	03		0512		BNE NOJ	
9361-	4C	82	94	0513		JMP JCOM	
9364-	C9	45		0514	NOJ	CMP #'E	
9366-	DO	03		0515		BNE NOE	
9368-	4C	A7	94	0516		JMP ECOM	
936B-	C9	54		0517	NOE	CMP #'T	
936D-	DO	03		0518		BNE NOT	
936F-	4C	B0	94	0519		JMP TCOM	
9372-	C9	2E		0520	NOT	CMP #'.	
9374-	DO	03		0521		BNE NOPER	
9376-	4C	B4	94	0522		JMP PERCOM	
9379-	C9	4F		0523	NOPER	CMP #'O	
937B-	DO	03		0524		BNE NOO	
937D-	4C	E6	94	0525		JMP OCOM	
9380-	C9	53		0526	NOO	CMP #'S	
9382-	DO	03		0527		BNE NOS	
9384-	4C	F8	94	0528		JMP SCOM	
9387-	C9	50		0529	NOS	CMP #'P	
9389-	DO	03		0530		BNE NOP	

=====

= Bij het ter perse gaan van deze editie bleek dat in =

= het Printer Facilities Program van het artikel Word =

= Processing with Carl Moser's Text Editor van auteur =

= Hans Buurman toch nog een kleine verbetering nodig =

= te zijn. =

= Regel 116 had moeten worden gevolgd door 4C 03 20 =

= JMP WARMST. =

= Deze kleinigheid kan echter snel verholpen worden =

= door de volgende wijziging: =

= 90A2 4C 83 9A JMP LABEL =

= 9A83 20 85 91 LABEL JSR SENDBUFS =

= 9A86 4C 03 20 JMP WARMST =

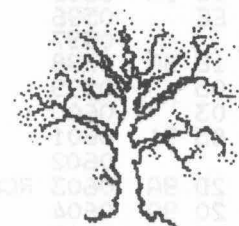
=====

= WIJ STAAN OP DE H.C.C.-COMPUTERDAGEN 16 EN 17 NOV 1984 =

= JAARBEURS UTRECHT =

= ONTMOET DAAR ONS 6502-TEAM =

=====



938B-	4C	0F	95	0531	JMP PCOM	
938E-	C9	2B		0532	CMP #' +	
9390-	D0	03		0533	BNE NOPLUS	
9392-	4C	24	95	0534	JMP PLUSCOM	
9395-	C9	42		0535	CMP #' B	
9397-	D0	03		0536	BNE NOB	
9399-	4C	2D	95	0537	JMP BCOM	
939C-	C9	41		0538	CMP #' A	
939E-	D0	03		0539	BNE NOA	
93A0-	4C	63	95	0540	JMP ACOM	
93A3-	C9	49		0541	CMP #' I	
93A5-	D0	03		0542	BNE NOI	
93A7-	4C	CC	96	0543	JMP ICOM	
93AA-	C9	23		0544	CMP #' #	
93AC-	D0	03		0545	BNE NORC	
93AE-	4C	8E	96	0546	JMP RCCOM	
93B1-	C9	46		0547	CMP #' F	
93B3-	D0	03		0548	BNE NOF	
93B5-	4C	42	96	0549	JMP FCOM	
93B8-	C9	4B		0550	CMP #' K	
93BA-	D0	03		0551	BNE NOK	
93BC-	4C	C6	96	0552	JMP KCOM	
93BF-	C9	43		0553	CMP #' C	
93C1-	D0	03		0554	BNE NOC	
93C3-	4C	08	97	0555	JMP CCOM	
93C6-	C9	51		0556	CMP #' Q	
93C8-	D0	03		0557	BNE NOQ	
93CA-	4C	1D	98	0558	JMP QCOM	
				0559	:UNKNOWN COMMAND: BC ERROR !	
93CD-	8D	28	01	0560	STA FILENR	
93D0-	A2	BC		0561	LDX #B3	
93D2-	4C	51	94	0562	JMP ERROR	
				0563	:COMCOM COMMON RETURN ADRESS FOR ALL COMMANDS	
93D5-	20	6B	90	0564	COMCOM JSR NEXT	
				0565	: MEANS OTHER COMMAND FOLLOWS	
93D8-	C9	3A		0566	CMP #' :	
93DA-	FO	07		0567	BEQ TOCOM	
				0568	: J MEANS END OF COMMANDS	
93DC-	C9	5D		0569	CMP #' J	
93DE-	D0	F5		0570	BNE COMCOM	
93E0-	4C	04	92	0571	JMP LOOP	
93E3-	4C	17	93	0572	JMP COMM	
				0573	:LCOM - LEFT BORDER COMMAND	
93E6-	20	D8	92	0574	LCOM JSR DECIM	
				0575	:MAY NOT BE LARGER THAN LINE LENGTH	
93E9-	CD	14	9A	0576	CMP HSIZE	
93EC-	BO	14		0577	BCS ERR30	
93EE-	8D	1F	9A	0578	STA LBORD	
				0579	: IF TO THE RIGHT OF PRESENT POSITION, CHANGE THE LATTER	
93F1-	CD	18	9A	0580	CMP HCNT	
93F4-	90	03		0581	BCC LCOM1	
93F6-	8D	18	9A	0582	STA HCNT	
				0583	:THERE MUST STILL BE ROOM FOR PARAGRAPH TABS	
93F9-	AD	21	9A	0584	LDA PTAB	
93FC-	8D	2D	9A	0585	STA NUMB	
93FF-	4C	85	94	0586	JMP JCOM1	
				0587	:ERROR 30 MESSAGE	
9402-	A2	30		0588	LDX #30	
9404-	8D	28	01	0589	STA FILENR	
9407-	4C	51	94	0590	JMP ERROR	
				0591	:RCOM - LEFT BORDER COMMAND	
940A-	20	D8	92	0592	RCOM JSR DECIM	
				0593	:MUST BE INSIDE THE LINE	
940D-	FO	F3		0594	BEQ ERR30	
940F-	CD	14	9A	0595	CMP HSIZE	
9412-	BO	EE		0596	BCS ERR30	
				0597	: IF TO THE LEFT OF CURRENT POSITION, START NEW LINE	
9414-	CD	18	9A	0598	CMP HCNT	
9417-	FO	05		0599	BEQ RCOM1	
9419-	BO	03		0600	BCS RCOM1	
941B-	20	85	91	0601	JSR SENDBUFS	
				0602	:THERE MUST BE ROOM FOR PARAGRAPH TABS	
941E-	AD	2D	9A	0603	LDA NUMB	
9421-	8D	20	9A	0604	STA RBORD	
9424-	4C	F9	93	0605	JMP LCOM1	
				0606	:DCOM - START NEW LINE	
9427-	20	85	91	0607	DCOM JSR SENDBUFS	

=====

ASM 65C 65CXXX-CROSS-ASSEMBLER V1.0
FOR SENIOR MONITOR
(= JUNIOR-como. with PROTON D.O.S.)

=====

The ASM 65C 65CXXX-CROSS-ASSEMBLER handles all new instructions of 65C02-processors and is fully compatible with 6502. Author: R. Banen. Schiedam. The Netherlands. Memory space needed: \$3000 - \$386A. Complete source listing available. English version: Fl. 52.50 incl. postage, packing, admin. etc. Foreign countries (CEPT countries excl.) Fl. 6.75 extra transfers.

To be paid to Mr. W.L. van Pelt, postal account nr. 841433 at Krimpen a.d. IJssel, or to bank-account nr. 44.11.06.471 of Mr. W.L. van Pelt AMRO-Bank at Krimpen a.d. IJssel. Eurocheque allowed.

=====

LAAT UW ZELF ONTWIKKELDE PROGRAMMA'S VOOR UW C-64.
APPLE, BBC, OSI, PET, ETC. ETC. NIET LIGGEN. MAAR
STUUR HET OP NAAR DE REDAKTIE. U HELPT ER NIET AL-
LEEN ANDEREN MEE. U HELPT HEN ER OOK MEE IDEEEN
TE ONTWIKKELEN DIE WEER GEPUBLICEERD KUNNEN WORD-
DEN. EN DAT IS UW BELANG !

=====

```

942A- 4C D5 93 0608      JMP COMCOM
                      0609 :HCOM - CHANGE HORIZONTAL MODE
942D- 20 D8 92 0610 HCOM      JSR DECIM
9430- 29 07 0611      AND #X111
9432- 20 11 91 0612      JSR HMODE
9435- 4C D5 93 0613      JMP COMCOM
                      0614 :VCOM - CHANGE VERTICAL MODE
9438- 20 D8 92 0615 VCOM      JSR DECIM
943B- 29 07 0616      AND #X111
943D- 20 3B 91 0617      JSR VMODE
9440- 4C D5 93 0618      JMP COMCOM
                      0619 :UCOM - UNDERLINE
9443- 20 D8 92 0620 UCOM      JSR DECIM
9446- 8D 12 9A 0621      STA UNDERL
9449- F0 03 0622      BEQ COMCOM2
944B- 8D 13 9A 0623      STA UNDERLD
944E- 4C D5 93 0624 COMCOM2    JMP COMCOM
                      0625 :ERROR - ISSUE ERROR MESSAGE
9451- A5 20 0626 ERROR      LDA *PNT
9453- 48 0627      PHA
9454- A5 21 0628      LDA *PNT+1
9456- 48 0629      PHA
                      0630 :MOVE BACK TO START OF LINE FOR LINE NUMBER
9457- 20 7B 92 0631 ERROR2    JSR LAST
945A- 90 FB 0632      BCC ERROR2
945C- 20 4B 24 0633      JSR ERRORM
                      0634 :CONTINUE
945F- 68 0635      PLA
9460- 85 21 0636      STA *PNT+1
9462- 68 0637      PLA
9463- 85 20 0638      STA *PNT
9465- 4C D5 93 0639      JMP COMCOM
                      0640 :C2COM - PRINT EXTRA DARK
9468- 8D 10 9A 0641 C2COM      STA USE2
946B- 8D 11 9A 0642      STA USED2
946E- 4C D5 93 0643      JMP COMCOM
                      0644 :C1COM - PRINT NORMAL
9471- A9 00 0645 C1COM      LDA #0
9473- 8D 10 9A 0646      STA USE2
9476- 4C D5 93 0647      JMP COMCOM
                      0648 :WCOM - TERMINAL WIDTH
9479- 20 D8 92 0649 WCOM      JSR DECIM
947C- 8D 22 9A 0650      STA TWIDTH
947F- 4C D5 93 0651      JMP COMCOM
                      0652 :JCOM - CHANGE PARAGRAPH TABULATION
9482- 20 D8 92 0653 JCOM      JSR DECIM
                      0654 :JCOM1 - CHECK TO SEE IF, WITH PARAGRAPH TAB, THE RESULTING
                      0655 :LINE IS STILL 16 CHARACTERS WIDE
                      0656 :LBORD+PTAB-RBORD<=-16
9485- 18 0657 JCOM1      CLC
9486- 6D 1F 9A 0658      ADC LBORD
9489- 38 0659      SEC
948A- ED 20 9A 0660      SBC RBORD
948D- B0 04 0661      BCS ERR31
948F- E9 F0 0662      SBC #F0
9491- 90 0B 0663      BCC JACC
                      0664 :LINE TOO SMALL, ERROR 31
9493- AD 2D 9A 0665 ERR31    LDA NUMB
9496- 8D 28 01 0666      STA FILENR
9499- A2 31 0667      LDX #31
949B- 4C 51 94 0668      JMP ERROR
                      0669 :PARAGRAPH TAB ACCEPTED
949E- AD 2D 9A 0670 JACC      LDA NUMB
94A1- 8D 21 9A 0671      STA PTAB
94A4- 4C D5 93 0672      JMP COMCOM
                      0673 :ECOM - ERROR CONTROL COMMAND
94A7- 20 D8 92 0674 ECOM      JSR DECIM
94AA- 8D 12 01 0675      STA CONTFI
94AD- 4C D5 93 0676      JMP COMCOM
                      0677 :TCOM - TABULATION COMMAND
94B0- A9 20 0678 TCOM      LDA #1
94B2- D0 02 0679      BNE PER1
                      0680 :PERCOM - TABULATE WITH PERIODS (.....)
94B4- A9 2E 0681 PERCOM     LDA #1
94B6- 8D 2E 9A 0682 PER1      STA TEMP
94B9- 20 D8 92 0683      JSR DECIM
                      0684 :TABULATE OUT OF THE LINE TO GET A 32 ERROR

```


94BC-	CD 20 9A	0685	CMP RBORD
94BF-	BO 1A	0686	BCS ERR32
		0687	:SEND THE SYMBOL IN TEMP UNTIL IN THE PROPER COLUMN
94C1-	AD 18 9A	0688	LDA HCNT
94C4-	CD 2D 9A	0689	CMP NUMB
94C7-	BO 09	0690	BCS PER3
94C9-	AD 2E 9A	0691	LDA TEMP
94CC-	20 37 92	0692	JSR PUT
94CF-	4C C1 94	0693	JMP PER2
		0694	:SUPPRESS SPACES AFTER A TAB
94D2-	AD 2E 9A	0695	LDA TEMP
94D5-	8D 2B 9A	0696	STA LCHAR
94D8-	4C D5 93	0697	JMP COMCOM
		0698	:SEND A 32 ERROR MESSAGE
94DB-	AD 2D 9A	0699	LDA NUMB
94DE-	8D 28 01	0700	STA FILENR
94E1-	A2 32	0701	LDX ##32
94E3-	4C 51 94	0702	JMP ERROR
		0703	:OCOM - OUTPUT CONTROL
94E6-	20 D8 92	0704	JSR DECIM
94E9-	48	0705	PHA
94EA-	29 01	0706	AND #*1
		0707	:BIT0 SWITCHES TERMINAL OUTPUT
94EC-	8D 1E 9A	0708	STA SCREEN
94EF-	68	0709	PLA
94F0-	29 02	0710	AND #*10
		0711	:BIT1 SWITCHES THE PRINTER
94F2-	8D 1D 9A	0712	STA PRINT
94F5-	20 D5 93	0713	JSR COMCOM
		0714	:SCOM - SEND A NUMBER OF SPACES
94F8-	20 D8 92	0715	JSR DECIM
94FB-	F0 0F	0716	BEQ SCOM2
		0717	:SEND NUMB SPACES
94FD-	A9 20	0718	LDA #'
94FF-	20 37 92	0719	JSR PUT
9502-	CE 2D 9A	0720	DEC NUMB
9505-	D0 F6	0721	BNE SCOM1
		0722	:SUPPRESS FOLLOWING SPACES
9507-	A9 20	0723	LDA #'
9509-	8D 2B 9A	0724	STA LCHAR
950C-	4C D5 93	0725	JMP COMCOM
		0726	:PCOM - START A NEW PARAGRAPH
950F-	20 B5 91	0727	JSR SENDBUFS
		0728	:TABULATE
9512-	AD 21 9A	0729	LDA PTAB
9515-	18	0730	CLC
9516-	6D 1F 9A	0731	ADC LBORD
9519-	8D 18 9A	0732	STA HCNT
		0733	:SUPPRESS SPACES
951C-	A9 20	0734	LDA #'
951E-	8D 2B 9A	0735	STA LCHAR
9521-	4C D5 93	0736	JMP COMCOM
		0737	:PLUSCOM - SET AN OFFSET
9524-	20 D8 92	0738	JSR DECIM
9527-	8D 23 9A	0739	STA OFFSET
952A-	4C D5 93	0740	JMP COMCOM
		0741	:BCOM - SEND TEXT TO SECOND BUFFER ONLY
952D-	AE 18 9A	0742	:USE POSITION IN TEMP, SECOND BUFFER
9530-	8E 2E 9A	0743	LDX HCNT
9533-	A2 FF	0744	STX TEMP
9535-	8E 11 9A	0745	LDX ##FF
		0746	STX USED2
		0747	:GET CHARACTER
9538-	20 6B 90	0748	JSR NEXT
		0749	:IS IT A TERMINATOR ?
953B-	C9 5D	0750	CMP #'J
953D-	F0 1E	0751	BEQ BCOMEND
953F-	C9 3A	0752	CMP #' :
9541-	F0 1A	0753	BEQ BCOMEND
		0754	:SEND TO SECOND BUFFER
9543-	AE 2E 9A	0755	LDX TEMP
9546-	9D 08 99	0756	STA BUF2.X
9549-	E8	0757	INX
954A-	EC 20 9A	0758	CPX RBORD
954D-	8E 2E 9A	0759	STX TEMP
		0760	:STOP IF END OF LINE REACHED
9550-	D0 E6	0761	BNE BCOM1

```

9552- 20 6B 90 0762 :THEN SKIP THE REST
9555- C9 5D 0763 BCOM2 JSR NEXT
9557- F0 04 0764 CMP #'J
9559- C9 3A 0765 BEQ BCOMEND
955B- D0 F5 0766 CMP #'I
955D- 20 7B 92 0767 BNE BCOM2
9560- 4C D5 93 0768 :TERMINATOR FOUND, MOVE BACK AND EXIT
0769 BCOMEND JSR LAST
0770 JMP COMCOM
0771 :ACOM - SEND AN ASCII-CODE
0772 ACOM JSR DECIM
0773 JSR PUT
0774 :SPACES MAY FOLLOW
0775 LDA #1
0776 STA LCHAR
0777 JMP COMCOM
0778 ;NEWLINE - AFTER A LINEFEED, CHECK FOR END OF PAGE
0779 :ADJUST POSITION COUNTER
9571- AD 15 9A 0780 NEWLINE LDA VSIZE
9574- 18 0781 CLC
9575- 6D 19 9A 0782 ADC VCNT
9578- 8D 19 9A 0783 STA VCNT
957B- AD 1A 9A 0784 LDA VCNT+1
957E- 69 00 0785 ADC #0
9580- 8D 1A 9A 0786 STA VCNT+1
0787 :CHECK FOR END OF PAGE
9583- 20 A7 95 0788 JSR PCHECK
9586- 90 1E 0789 BCC ENDNL
0790 ;FFCOM - START A NEW PAGE
0791 :RESET COUNTER
9588- A9 00 0792 FFCOM LDA #0
958A- 8D 19 9A 0793 STA VCNT
958D- 8D 1A 9A 0794 STA VCNT+1
9590- AD 1B 9A 0795 LDA PAGED
9593- F0 11 0796 BEQ ENDNL
0797 :PAGES USED : ADJUST PAGE NR
9595- AD 1C 9A 0798 LDA PAGE
9598- 18 0799 CLC
9599- F8 0800 SED
959A- 69 01 0801 ADC #1
959C- D8 0802 CLD
959D- 8D 1C 9A 0803 STA PAGE
0804 :INSERT OPEN LINES AND HEADING
95A0- 20 B6 95 0805 JSR BETP
95A3- 20 FC 95 0806 JSR PAGEHEAD
95A6- 60 0807 ENDNL RTS
0808 :PCHECK - COMPARE VCNT AND PSIZE
95A7- AD 1A 9A 0809 PCHECK LDA VCNT+1
95AA- CD 25 9A 0810 CMP PSIZE+1
95AD- D0 06 0811 BNE PCH1
95AF- AD 19 9A 0812 LDA VCNT
95B2- CD 24 9A 0813 CMP PSIZE
95B5- 60 0814 PCH1 RTS
0815 :BETP - SPACE BETWEEN PAGES
95B6- AD 26 9A 0816 BETP LDA BSIZE
95B9- C9 FF 0817 CMP #255
95BB- D0 04 0818 BNE BETP1
0819 :BSIZE=255 : JUST WAIT FOR A KEYSTROKE
95BD- 20 AE 12 0820 JSR RECHA
95C0- 60 0821 RTS
0822 :LINES NEEDED: SAVE PRINTER MODE
95C1- AD 17 9A 0823 BETP1 LDA VM
95C4- 48 0824 PHA
0825 :BSIZE>127 : MODE 1 ELSE MODE 0
95C5- AD 26 9A 0826 LDA BSIZE
95C8- 30 04 0827 BMI MO1
95CA- A9 00 0828 LDA #0
95CC- F0 02 0829 BEQ BETP2
95CE- A9 01 0830 MO1 LDA #1
95D0- 20 3B 91 0831 BETP2 JSR VMODE
0832 :SEND DESIRED NUMBER OF LINEFEEDS. TEMP+1 IS COUNTER
95D3- AD 26 9A 0833 LDA BSIZE
95D6- 29 7F 0834 AND #7F
95D8- 8D 2F 9A 0835 STA TEMP+1
95DB- F0 1A 0836 BETP3 BEQ BETP4
0837 :SEND TO SELECTED OUTPUTS, BUT DON'T CHECK FOR PAGES
95DD- AD 1E 9A 0838 LDA SCREEN

```


95E0-	FO 05	0839	BEQ BETP3B
95E2-	A9 0A	0840	LDA ##A
95E4-	20 80 0C	0841	JSR OUTCH
95E7-	AD 1D 9A	0842	LDA PRINT
95EA-	FO 05	0843	BEQ BETP3A
95EC-	A9 0A	0844	LDA ##A
95EE-	20 0C 40	0845	JSR PRINTER
95F1-	CE 2F 9A	0846	DEC TEMP+1
95F4-	4C DB 95	0847	JMP BETP3
		0848	:RESTORE PRINTER MODE. EXIT
95F7-	68	0849	PLA
		0850	.LI WORDPR.M03

F03 1CE6 4292-5F78

95F8-	20 3B 91	0850	JSR VMODE
95FB-	60	0851	RTS
		0852	:PAGEHEAD - PRINT PAGE HEADING (WITH LINE NR)
		0853	:FIRST: MOVE TO HSIZE/2-1 (THE MIDDLE)
95FC-	AD 14 9A	0854	LDA HSIZE
95FF-	4A	0855	LSR A
9600-	8D 2F 9A	0856	STA TEMP+1
9603-	CE 2F 9A	0857	DEC TEMP+1
		0858	:SEND ENOUGH SPACES
9606-	A9 20	0859	LDA #'
9608-	20 3C 90	0860	JSR OUT
960B-	CE 2F 9A	0861	DEC TEMP+1
960E-	D0 F6	0862	BNE PAG1
		0863	:GET PAGE NR HIGH NIBBLE
9610-	AD 1C 9A	0864	LDA PAGE
9613-	4A	0865	LSR A
9614-	4A	0866	LSR A
9615-	4A	0867	LSR A
9616-	4A	0868	LSR A
		0869	:IF ZERO. PRINT A SPACE
9617-	D0 04	0870	BNE PAG2A
9619-	A9 20	0871	LDA #'
961B-	D0 03	0872	BNE PAG2B
961D-	18	0873	CLC
961E-	69 30	0874	ADC ##30
9620-	20 3C 90	0875	JSR OUT
		0876	:PRINT CR AND THREE LINEFEEDS
9623-	AD 1C 9A	0877	LDA PAGE
9626-	29 0F	0878	AND ##F
9628-	09 30	0879	ORA ##110000
962A-	20 3C 90	0880	JSR OUT
962D-	A9 0D	0881	LDA ##D
962F-	20 3C 90	0882	JSR OUT
9632-	A9 03	0883	LDA #3
9634-	8D 2F 9A	0884	STA TEMP+1
9637-	A9 0A	0885	LDA ##A
9639-	20 38 90	0886	JSR OUT0
963C-	CE 2F 9A	0887	DEC TEMP+1
963F-	D0 F6	0888	BNE PAG3
		0889	:NOTE: TEMP+1 IS 00 AFTER A FORM FEED
9641-	60	0890	RTS
		0891	:FCOM - EJECT TO TOP OF NEXT PAGE. SEE .EJ PSEUDO OP
		0892	: IF NOT USED BEFORE, JUST START NEW PAGE
9642-	AD 1B 9A	0893	LDA PAGED
9645-	FO 3C	0894	BEQ PAGON
		0895	:TEMP+1 (>)0 .WILL BE 0 AFTER FORM FEED
9647-	8D 2F 9A	0896	STA TEMP+1
964A-	20 85 91	0897	JSR SENDBUFS
		0898	:SAVE PRINTER MODE
964D-	AD 17 9A	0899	LDA VM
9650-	48	0900	PHA
		0901	:MODE 1 UNTIL A MULTIPLE OF 8/48 INCH REACHED
9651-	A9 01	0902	LDA #1
9653-	20 3B 91	0903	JSR VMODE
		0904	:IS IT REACHED ?
9656-	AD 19 9A	0905	LDA VCNT
9659-	29 07	0906	AND ##111
965B-	FO 0D	0907	BEQ FCOM2
		0908	:TEST IF FF ALREADY GIVEN
965D-	AD 2F 9A	0909	LDA TEMP+1
9660-	FO 1A	0910	BEQ ENDFCOM
		0911	:NO. SEND LF AND CONTINUE

9662-	A9 0A	0912	LDA ##A
9664-	20 38 90	0913	JSR OUTO
9667-	4C 56 96	0914	JMP FCOM1
		0915	:NOW TRY MODE 0
966A-	A9 00	0916 FCOM2	LDA #0
966C-	20 3B 91	0917	JSR VMODE
		0918	:TEST IF FF ALREADY GIVEN
966F-	AD 2F 9A	0919 FCOM2B	LDA TEMP+1
9672-	F0 08	0920	BEQ ENDFCOM
		0921	:NO, SEND LINEFEEDS
9674-	A9 0A	0922	LDA ##A
9676-	20 38 90	0923	JSR OUTO
9679-	4C 6F 96	0924	JMP FCOM2B
		0925	:FF GIVEN, RESTORE PRINTER MODE AND EXIT
967C-	68	0926 ENDFCOM	PLA
967D-	20 3B 91	0927	JSR VMODE
9680-	4C D5 93	0928	JMP COMCOM
		0929	:PAGON - BEGIN PAGED MODE, START NEW PAGE
9683-	A9 FF	0930 PAGON	LDA ##FF
9685-	8D 1B 9A	0931	STA PAGED
9688-	20 88 95	0932	JSR FFCOM
968B-	4C D5 93	0933	JMP COMCOM
		0934	:RCCOM - READ A DECIMAL PAGE NUMBER
968E-	A9 00	0935 RCCOM	LDA #0
9690-	8D 1C 9A	0936	STA PAGE
9693-	20 6B 90	0937 RC1	JSR NEXT
		0938	:TEST FOR NUMBER
9696-	20 88 96	0939	JSR NUMTEST
9699-	30 17	0940	BMI RC2
		0941	:IT IS, SHIFT IT IN
969B-	0E 1C 9A	0942	ASL PAGE
969E-	0E 1C 9A	0943	ASL PAGE
96A1-	0E 1C 9A	0944	ASL PAGE
96A4-	0E 1C 9A	0945	ASL PAGE
96A7-	29 0F	0946	AND ##F
96A9-	0D 1C 9A	0947	ORA PAGE
96AC-	8D 1C 9A	0948	STA PAGE
96AF-	4C 93 96	0949	JMP RC1
		0950	:IT ISN'T, EXIT
96B2-	20 7B 92	0951 RC2	JSR LAST
96B5-	4C D5 93	0952	JMP COMCOM
		0953	:NUMTEST - SET N=1 IF NO NUMBER, ELSE N=0
96B8-	C9 30	0954 NUMTEST	CMP ##30
96BA-	30 07	0955	BMI NUMT1
96BC-	C9 3A	0956	CMP ##3A
96BE-	10 03	0957	BPL NUMT1
96C0-	A0 00	0958	LDY #0
96C2-	60	0959	RTS
96C3-	A0 FF	0960 NUMT1	LDY ##FF
96C5-	60	0961	RTS
		0962	:KCOM - WAIT FOR ANY KEY
96C6-	20 AE 12	0963 KCOM	JSR RECHA
96C9-	4C D5 93	0964	JMP COMCOM
		0965	:ICOM - SET PAGE SIZE
96CC-	20 D8 92	0966 ICOM	JSR DECIM
96CF-	D0 06	0967	BNE ICOM2
		0968	:FIRST OPERAND 0, SWITCH PAGED OFF AND EXIT
96D1-	8D 1B 9A	0969	STA PAGED
96D4-	4C D5 93	0970	JMP COMCOM
		0971	:FIRST OPERAND IS SIZE IN INCHES, MULTIPLY BY 48
96D7-	A2 00	0972 ICOM2	LDX #0
96D9-	8E 25 9A	0973	STX PSIZE+1
		0974	:FIRST BY 3
96DC-	8D 24 9A	0975	STA PSIZE
96DF-	0A	0976	ASL A
96E0-	18	0977	CLC
96E1-	6D 24 9A	0978	ADC PSIZE
96E4-	8D 24 9A	0979	STA PSIZE
		0980	:THEN 4 TIMES BY 2 (RESULT: BY 16)
96E7-	A2 04	0981	LDX ##4
96E9-	0E 24 9A	0982 ICOM3	ASL PSIZE
96EC-	2E 25 9A	0983	ROL PSIZE+1
96EF-	CA	0984	DEX
96F0-	D0 F7	0985	BNE ICOM3
		0986	:TOTAL RESULT: BY 48
		0987	:NOW CHECK IF SECOND OPERAND FOLLOWS
96F2-	20 6B 90	0988	JSR NEXT

96F5-	C9 2C	0989	CMP #'
96F7-	D0 09	0990	BNE ICOM4
		0991	:IT DOES, JUST READ IT IN
96F9-	20 D8 92	0992	JSR DECIM
96FC-	8D 26 9A	0993	STA BSIZE
96FF-	4C D5 93	0994	JMP COMCOM
		0995	:EXIT FROM ICOM
9702-	20 7B 92	0996	JSR LAST
9705-	4C D5 93	0997	JMP COMCOM
		0998	:CCOM - CHANGE LINEFEEDS AFTER EACH PASS
9708-	20 D8 92	0999	CCOM JSR DECIM
970B-	29 03	1000	AND #11
970D-	8D 2E 9A	1001	STA TEMP
9710-	D0 08	1002	BNE CCOM1
		1003	:FIRST OPERAND 0. ERROR 33
9712-	8D 28 01	1004	ERR33 STA FILENR
9715-	A2 33	1005	LDX #33
9717-	4C 51 94	1006	JMP ERROR
971A-	20 6B 90	1007	CCOM1 JSR NEXT
		1008	:IF NO SECOND OPERAND. ERROR 33
971D-	C9 2C	1009	CMP #'
971F-	D0 F1	1010	BNE ERR33
9721-	20 D8 92	1011	JSR DECIM
		1012	:BIT 0 DETERMINES LINEFEED OR NOT
9724-	29 01	1013	AND #1
9726-	F0 02	1014	BEQ CCOM2
9728-	A9 0A	1015	LDA #A
972A-	AE 2E 9A	1016	CCOM2 LDX TEMP
972D-	9D 26 9A	1017	STA CTRL-1.X
9730-	4C D5 93	1018	JMP COMCOM
		1019	:DIVIDE - QUOT:=INT(SPACES/WORDS)
		1020	: QUOT:=0:
		1021	: WHILE SPACES >= WORDS DO
		1022	: QUOT:=QUOT+1: SPACES:=SPACES-WORDS:
9733-	A9 00	1023	DIVIDE LDA #0
9735-	8D 32 9A	1024	STA QUOT
9738-	AD 31 9A	1025	DIV1 LDA SPACES
973B-	CD 30 9A	1026	CMP WORDS
973E-	90 0C	1027	BCC ENDDIV
9740-	ED 30 9A	1028	SBC WORDS
9743-	8D 31 9A	1029	STA SPACES
9746-	EE 32 9A	1030	INC QUOT
9749-	4C 38 97	1031	JMP DIV1
974C-	60	1032	ENDDIV RTS
		1033	:FILLOUT - SPREAD WORDS OVER THE LINE
		1034	:FIRST COUNT SPACES AND WORDS-1
974D-	A2 00	1035	FILLOUT LDX #0
974F-	8E 31 9A	1036	STX SPACES
9752-	CA	1037	DEX
9753-	8E 30 9A	1038	STX WORDS
		1039	:SAVE HCNT. USE HCNT AS COUNTER
9756-	AD 18 9A	1040	LDA HCNT
9759-	48	1041	PHA
975A-	AD 1F 9A	1042	LDA LBORD
975D-	8D 18 9A	1043	STA HCNT
9760-	AC 20 9A	1044	LDY RBORD
		1045	:FIND FIRST WORD. COUNT SPACES
9763-	AE 18 9A	1046	FILL1 LDX HCNT
9766-	BD 84 98	1047	LDA BUF1.X
9769-	C9 20	1048	CMP #'
976B-	D0 0D	1049	BNE FILL2
976D-	EE 18 9A	1050	INC HCNT
9770-	EE 31 9A	1051	INC SPACES
9773-	CC 18 9A	1052	CPY HCNT
9776-	D0 EB	1053	BNE FILL1
9778-	F0 2C	1054	BEQ ENDP1
		1055	:WORD FOUND. COUNT AND SKIP IT
977A-	EE 30 9A	1056	FILL2 INC WORDS
977D-	EE 18 9A	1057	FILL3 INC HCNT
9780-	CC 18 9A	1058	CPY HCNT
9783-	F0 21	1059	BEQ ENDP1
9785-	AE 18 9A	1060	LDX HCNT
9788-	BD 84 98	1061	LDA BUF1.X
978B-	C9 20	1062	CMP #'
978D-	D0 EE	1063	BNE FILL3
		1064	:SPACES FOUND. COUNT AND SKIP THEM
978F-	EE 31 9A	1065	FILL4 INC SPACES

9792-	EE	18	9A	1066	INC HCNT
9795-	CC	18	9A	1067	CPY HCNT
9798-	FO	0C		1068	BEQ ENDP1
979A-	AE	18	9A	1069	LDA HCNT
979D-	BD	84	98	1070	LDA BUF1,X
97A0-	C9	20		1071	CMP #'
97A2-	FO	EB		1072	BEQ FILL4
97A4-	DO	D4		1073	BNE FILL2
				1074	:END OF LINE REACHED, COUNTING DONE
97A6-	AD	30	9A	1075	LDA WORDS
97A9-	30	6D		1076	BMI ENDP2
				1077	:ENOUGH WORDS TO FILL OUT ?
97AB-	CD	2A	9A	1078	CMP FILL
97AE-	30	68		1079	BMI ENDP2
97B0-	20	33	97	1080	JSR DIVIDE
				1081	:YES, DIVIDE AND START FILLING OUT
97B3-	AD	20	9A	1082	LDA RBORD
97B6-	8D	18	9A	1083	STA HCNT
97B9-	AB			1084	TAY
				1085	:MOVE FROM BUF,X TO BUF,Y (X FROM HCNT)
				1086	:FIND LAST WORD
97BA-	AE	18	9A	1087	LDA HCNT
97BD-	EC	1F	9A	1088	CPX LBORD
97C0-	FO	56		1089	BEQ ENDP2
97C2-	CE	18	9A	1090	DEC HCNT
97C5-	CA			1091	DEX
97C6-	BD	84	98	1092	LDA BUF1,X
97C9-	C9	20		1093	CMP #'
97CB-	FO	ED		1094	BEQ FILL5
97CD-	88			1095	DEY
				1096	:WORD FOUND, MOVE IT
97CE-	BD	84	98	1097	LDA BUF1,X
97D1-	C9	20		1098	CMP #'
97D3-	FO	19		1099	BEQ FILL7
97D5-	99	84	98	1100	STA BUF1,Y
97D8-	BD	08	99	1101	LDA BUF2,X
97DB-	99	08	99	1102	STA BUF2,Y
97DE-	BD	8C	99	1103	LDA BUF3,X
97E1-	99	8C	99	1104	STA BUF3,Y
97E4-	EC	1F	9A	1105	CPX LBORD
97E7-	FO	2F		1106	BEQ ENDP2
97E9-	88			1107	DEY
97EA-	CA			1108	DEX
97EB-	4C	CE	97	1109	JMP FILL6
				1110	:SPACES FOUND, INSERT RIGHT NUMBER AND SKIP THE REST
97EE-	8E	18	9A	1111	STX HCNT
				1112	:COMPUTE RIGHT NUMBER
97F1-	AD	31	9A	1113	LDA SPACES
97F4-	FO	05		1114	BEQ FILL8
97F6-	A9	01		1115	LDA #1
97F8-	CE	31	9A	1116	DEC SPACES
97FB-	18			1117	CLC
97FC-	6D	32	9A	1118	ADC QUOT
97FF-	AA			1119	TAX
				1120	:INSERT SPACES
9800-	A9	20		1121	LDA #'
9802-	99	84	98	1122	STA BUF1,Y
9805-	99	08	99	1123	STA BUF2,Y
9808-	99	8C	99	1124	STA BUF3,Y
980B-	88			1125	DEY
980C-	CA			1126	DEX
980D-	DO	F3		1127	BNE FILL9
980F-	CE	18	9A	1128	DEC HCNT
9812-	AE	18	9A	1129	LDA HCNT
9815-	4C	CE	97	1130	JMP FILL6
				1131	:END OF FILLING OUT, RETSTORE HCNT
9818-	68			1132	PLA
9819-	8D	18	9A	1133	STA HCNT
981C-	60			1134	RTS
				1135	:QCOM - SET MINIMAL NUMBER OF WORDS TO FILL OUT
981D-	20	D8	92	1136	JSR DECIM
9820-	8D	2A	9A	1137	STA FILL
9823-	4C	D5	93	1138	JMP COMCOM
				1139	:SWEEP - REMOVE SUPERFLUOUS SPACES
				1140	:QUOT IS USED TO HOLD PREVIOUS CHARACTER
9826-	AE	1F	9A	1141	LDX LBORD
9829-	AC	1F	9A	1142	LDY LBORD

982C-	A9	20	1143	:FIND FIRST WORD
982E-	8D	32	9A	1144 LDA #'
9831-	BD	34	98	1145 STA QUOT
9834-	C9	20		1146 LDA BUF1.X
9836-	FO	20		1147 CMP #'
			1148 BEQ SW2	
			1149 :WORD: MOVE IT	
9838-	99	84	98	1150 SW12 STA BUF1.Y
983B-	8D	32	9A	1151 STA QUOT
983E-	BD	08	99	1152 LDA BUF2.X
9841-	99	08	99	1153 STA BUF2.Y
9844-	BD	8C	99	1154 LDA BUF3.X
9847-	99	8C	99	1155 STA BUF3.Y
984A-	C8			1156 INY
984B-	CC	20	9A	1157 CPY RBORD
984E-	FO	1D		1158 BEQ ENDSW
9850-	E8			1159 INX
9851-	EC	20	9A	1160 CPX RBORD
9854-	DO	DB		1161 BNE SW1
9856-	FO	15		1162 BEQ ENDSW
			1163 :SPACE: MOVE IF PREVIOUS WASN'T. ELSE REMOVE	
9858-	AD	32	9A	1164 SW2 LDA QUOT
985B-	C9	20		1165 CMP #'
985D-	DO	08		1166 BNE SW11
985F-	E8			1167 INX
9860-	EC	20	9A	1168 CPX RBORD
9863-	DO	CC		1169 BNE SW1
9865-	FO	06		1170 BEQ ENDSW
			1171 :MOVE SPACE	
9867-	BD	84	98	1172 SW11 LDA BUF1.X
986A-	4C	38	98	1173 JMP SW12
			1174 :ADD TRAILING SPACES	
986D-	A9	20		1175 ENDSW LDA #'
986F-	CC	20	9A	1176 CPY RBORD
9872-	FO	0F		1177 BEQ ENDSW2
9874-	99	84	98	1178 STA BUF1.Y
9877-	99	08	99	1179 STA BUF2.Y
987A-	99	8C	99	1180 STA BUF3.Y
987D-	C8			1181 INY
987E-	CC	20	9A	1182 CPY RBORD
9881-	DO	EA		1183 BNE ENDSW
9883-	60			1184 ENDSW2 RTS
			1185 :VARIABLES:	
9884-			1186 BUF1	.DS BUFLEN :FIRST BUFFER
9908-			1187 BUF2	.DS BUFLEN :SECOND BUFFER
998C-			1188 BUF3	.DS BUFLEN :UNDERLINE BUFFER
9A10-			1189 USE2	.DS 1 :FLAG : COPY TO BUF2
9A11-			1190 USED2	.DS 1 :FLAG : BUF2 USED
9A12-			1191 UNDERL	.DS 1 :FLAG : UNDERLINE IN BUF3
9A13-			1192 UNDERLD	.DS 1 :FLAG : BUF3 USED
9A14-			1193 HSIZE	.DS 1 :HORIZONTAL SIZE
9A15-			1194 VSIZE	.DS 1 :VERTICAL SIZE
9A16-			1195 HM	.DS 1 :HORIZONTAL MODE
9A17-			1196 VM	.DS 1 :VERTICAL MODE
9A18-			1197 HCNT	.DS 1 :HORIZONTAL POSITION
9A19-			1198 VCNT	.DS 2 :SAME, BUT VERTICAL
9A1B-			1199 PAGED	.DS 1 :FLAG : PAGED MODE
9A1C-			1200 PAGE	.DS 1 :PAGE COUNTER
9A1D-			1201 PRINT	.DS 1 :FLAG : OUTPUT TO PRINTER
9A1E-			1202 SCREEN	.DS 1 :FLAG : OUTPUT TO SCREEN
9A1F-			1203 LBORD	.DS 1 :LEFT BORDER
9A20-			1204 RBORD	.DS 1 :RIGHT BORDER
9A21-			1205 PTAB	.DS 1 :PARAGRAPH TABULATION
9A22-			1206 TWIDTH	.DS 1 :TERMINAL WIDTH
9A23-			1207 OFFSET	.DS 1 :NUMBER ADDED TO DATA
9A24-			1208 PSIZE	.DS 2 :SIZE OF PAGE IN 1/48 INCH
9A26-			1209 BSIZE	.DS 1 :SIZE OF SPACE BETWEEN PAGES
9A27-			1210 CTRL	.DS 3 :PRINTER CONTROL CODES
9A2A-			1211 FILL	.DS 1 :FLAG : FILLING OUT LINES
9A2B-			1212 LCHAR	.DS 1 :LAST CHAR SENT TO BUF1
9A2C-			1213 CCHAR	.DS 1 :CHAR FETCHED BY NEXT OR END OF LINE
9A2D-			1214 NUMB	.DS 1 :BUFFER FOR DECIMAL INPUT
9A2E-			1215 TEMP	.DS 2 :TEMPORARY STORAGE
9A30-			1216 WORDS	.DS 1 :NUMBER OF WORDS-1 IN BUF1
9A31-			1217 SPACES	.DS 1 :NUMBER OF SPACES IN BUF1
9A32-			1218 QUOT	.DS 1 :SPACES/WORDS
			1219 :PRINTER DEPENDENT PART	


```

1220 :OKI MICROLINE 80 FORMATS
1221 :1ST : HORIZONTAL SIZES
1222 :FORMAT : .BY X X X X Y WHERE XS ARE SENT TO THE PRINTER
1223 :AND Y IS NUMBER OF CHARACTERS/LINE
1224 :EIGHT MODES MAY BE USED (0...7)
9A33- 1B 41 1D 1225 HOR .BY $1B 'A' $1D 0 132 :COMPRESSED LONG LINE
9A36- 00 84
9A38- 1B 41 1E 1226 .BY $1B 'A' $1E 0 80 :NORMAL
9A3B- 00 50
9A3D- 1B 41 1F 1227 .BY $1B 'A' $1F 0 40 :DOUBLE WIDTH
9A40- 00 28
9A42- 1B 42 1D 1228 .BY $1B 'B' $1D 0 105 :SAME BUT SHORT LINE
9A45- 00 69
9A47- 1B 42 1E 1229 .BY $1B 'B' $1E 0 64
9A4A- 00 40
9A4C- 1B 42 1F 1230 .BY $1B 'B' $1F 0 32
9A4F- 00 20
9A51- 1B 42 1F 1231 .BY $1B 'B' $1F 0 32
9A54- 00 20
9A56- 1B 42 1F 1232 .BY $1B 'B' $1F 0 32
9A59- 00 20
1233 :2ND : VERTICAL
1234 :SAME FORMAT, BUT Y IS HEIGHT OF LINE TIMES 1/48 INCH
9A5B- 1B 36 00 1235 VER .BY $1B '6' 0 0 8 :6 LINES/INCH
9A5E- 00 08
9A60- 1B 38 00 1236 .BY $1B '8' 0 0 6 :8 LINES/INCH
9A63- 00 06
9A65- 1B 38 00 1237 .BY $1B '8' 0 0 6
9A68- 00 06
9A6A- 1B 38 00 1238 .BY $1B '8' 0 0 6
9A6D- 00 06
9A6F- 1B 38 00 1239 .BY $1B '8' 0 0 6
9A72- 00 06
9A74- 1B 38 00 1240 .BY $1B '8' 0 0 6
9A77- 00 06
9A79- 1B 38 00 1241 .BY $1B '8' 0 0 6
9A7C- 00 06
9A7E- 1B 38 00 1242 .BY $1B '8' 0 0 6
9A81- 00 06
1243 :WHEN CHANGING TO ANOTHER PRINTER, DON'T FORGET TO ADJUST BUFLIN !!!!
1244 :WRITTEN BY J.BUURMAN
1245 : v.d.FUYCKSTRAAT 136
1246 : 3232 AS BRIELLE (01810-5153)
1247 :LAST MODIFICATION 19-8-1984
1248 :.EN

```

AANGEBODEN:

KIM-1 met 64K Ram (Elektuurkaart). Hazeltine 1400 Video-terminal. Eventueel voeding voor KIM-1 (5V, + en -, 12V + en -, 25V). ASCII keyboard in behuizing.
KIM-1 en terminal in een kooi, met boeken.
Robert Korving, W. Pijperstraat 6. 2551 Den Haag.
Tel.: 070 - 680146 (tussen 19.00 en 21.00 uur).

KLEINTJE FORTH. REACTIES.

Ronald Vinks oplossing voor het listen van screens 0 t/m 15 in editie 33 van DE 6502 KENNER o. 47 vroeg om een oplossing voor het variabel listen van screens, op te nemen in het vocabulary.
Wout van Dinther. Molenrijnselaan 27. 5262 TN Vught, heeft aan de oproep gevolg gegeven. Hier is zijn oplossing.

```
: GLIST 1+ SWAP DO KEY I LIST LOOP :
```

Met als definitie omschrijving (n1 n2 --), waarbij n1 het eerste screen en n2 het laatste moet zijn. De KEY erin zorgt ervoor dat hij op ingave van het toetsenbord staat te wachten (b.v.: soatie), waarna het volgende screen gelist wordt.
Wout denkt inmiddels na over een nog mooiere oplossing. De redactie houdt zich aanbevolen voor meer FORTH-screens.

Fridus Jonkman oaf in Kleintje Fortn voor het Recursie-probleem. Er zijn meer oplossingen denkbaar. Een ervan ontvinden we van Gert van Oobroek. Hoollanden 20. 9801 LB Zuidhorn (Gr). Hij schreef ons het volgende:
Recursie kan op zeer eenvoudige wijze gedaan worden d.m.v. de definitie:

```
: RECURSIE ." 6502 " R) DROP ( zorg ervoor dat de return-  
 ( stack niet volloopt )
```

MYSELF :

Een ander voorbeeld:

```
: RECFAC ( bereken n! recursief )  
DUP 1 ) IF DUP 1- MYSELF * ENDIF :
```

Aanroep : n RECFAC .

65816 MICROPROCESSOR

Wie mocht denken dat de 65XX-familie het moeilijk heeft in de concurrentiestrijd, komt van een koude kermis terug. De 65XX-familie baart nog steeds nieuwe chios. De 65816 microprocessor is zo'n nieuwe boreling, intern uitgerust met een 16 bits accumulator, 16 bits indexregister, een 24 bits (!) programcounter en een 16 bits stackpointer. Naar analogie van de 6809 een 'direct page register', waardoor zero page in elk deel v/h geheugen te plaatsen. Met behoud van compatibiliteit met de 6502 als 65C02 te gebruiken, en de mogelijkheid om 16 mega(!)byte te adresseren. En nog heel veel meer van dat leuks. We hopen hier binnenkort nog uitbreider over te kunnen berichten. Bedankt inzenders !

UITNODIGING BIJEENKOMST

Datum : zaterdag 24 november 1984
 Lokatie : R.K. HTS "Rijswijk"
 Lange Kleiweg 4 te Rijswijk. Tel.: 070 - 907839

Reisroute :

- per auto - komende uit de richting Utrecht
 Volg autoweg E8 Utrecht-Den Haag. Knooppunt Leidschendam via de hoge
 rijbaan linksaf richting Delft/Rotterdam. Aan het eind daarvan rechts-
 af richting Den Haag. Rechts aanhouden. Onder verkeersbord Yoenburg
 door en recht door met trambaan mee over Hoornbrug. Dan scherpe draai
 naar rechts en onder brug door. Delft/Rotterdam en Wateringen aan-
 houden. Bij stoplichten linksaf en Sir Winston Churchillaan uitrijden
 tot laatste stoplichten voor spoorovergang. Links van U gebouw GAK en
 ABN bank. hier linksaf Huis te Landelaan oo. uitrijden tot eind en
 rechtsaf. Utrijden tot schoolleijn.
- komende uit de richting Amsterdam
 Volg A4 A'dam-Rotterdam. Knooppunt Leidschendam passeren. E8/E10 aan-
 houden tot eind. rechtsaf richting Den Haag. Rechts aanhouden. Onder
 verkeersbord Yoenburg door met trambaan mee over Hoornbrug. Verder
 als boven.
- komende uit de richting Rotterdam
 Volg E10 R'dam-Den Haag tot vlak voor Hoornbrug. Onder verkeersbord
 Yoenburg door en recht door met trambaan mee over Hoornbrug. Verder
 als boven.
- per trein - Station Rijswijk uitgaand richting winkelcentrum De Boogaard. Bij uit-
 gang linksaf. Spoorovergang Winston Churchillaan over tot stoplicht.
 rechtsaf Huis te Landelaan. Verder als boven.

Lunchoakket zelf meenemen.
 Consumpties tegen betaling.

Programma:

- 10.00 Opening Ledenvergadering 1984.
- 10.15 - Notulen jaarvergadering 1983
- Financieel Jaarverslag 1983
- Begroting 1984
- Verkiezing kascontrolecommissie
- Verkiezing bestuursleden:
 - aftredend en niet herkiesbaar: Ruud Uthoff
 - voordracht van het bestuur : Gert Klein
 - aftredend en niet herkiesbaar: Mw. M.F. Uthoff-Burghoorn
 - verkiezing 1 bestuurslid (kandidaten kunnen schriftelijk wor-
 den aangemeld bij de sekretaris of mondeling voor de aanvang
 van de vergadering).
 - Mededeling over c.a. uitreiking publiekatie-aanmoedigingspremie(s)
 - Rondvraag en sluiting.
- 12.00 Lunch
- 13.00 Lezing Nico de Vries over universeel programmeren. 65C02. enz.
- 14.00 INFORMEEL GEDEELTE
- BRENG OOK UW SYSTEEM MEE ALS DAT ENIGSZINS LUKT !!
- MARKT. Op eigen tafel(s) te regelen.
- 17.00 Sluiting.

```

100 // *****
110 // *
120 // * PROGRAMMA STAAFDIAGRAM *
130 // * voor microline 80 printer *
140 // *
150 // *****

```

Staafdiagram afdrukken met een microline 80 printer

Dit programma is geschreven om de resultaten van proefwerken van verschillende klassen met elkaar te vergelijken. Maar er zijn meer mogelijkheden te bedenken. Het programma is een voorbeeld om het gebruik van globale en lokale variabelen te demonstreren. (PROC "voer het aantal in"). Van de andere kant is het in COMAL*KGN niet mogelijk om karakters met een nummer groter dan 127 in een print statement te gebruiken. Helaas vormen de karakters boven de 127 juist de grafische set van de microline 80 printer. De oplossing is een in COMAL geschreven print-routine die het gewenste karakter direct in de printer oort "pookt". Zie de procedure "print" onderaan. Om het programma aanpasbaar te maken heb ik de poorten bovenaan in het programma gedeclareerd. Het zijn de variabelen:

STATUS de inputpoort van de PIA waarin de printer status staat (P7=online/offline : P6=paper 1=ot of paper : P5=BUSY : P4=not fault : P0-P3 =1)
 PIAPB de outputpoort naar de printer (parallel)
 ONLINE de waarde die de statuspoort heeft als de printer "busy" is.

Er kan in het programma gekozen worden : de resultaten afdrukken zoals ze zijn, of de resultaten in procenten afdrukken. In het eerste geval worden er getallen voor de regels afgedrukt. In het tweede geval geven streepjes een verdeling aan. Een voorbeeld is bijgevoegd. De getallen onderaan geven het nummer van de vraag.

```

100 // *****
101 // *
102 // * PROGRAMMA STAAFDIAGRAM *
103 // * voor microline 80 printer *
104 // * frans bakx *
105 // * huissteden 1112 *
106 // * 6605 hd wijchen *
107 // *
108 // * COMAL*KGN *
109 // *****
110 CLEAR
111 STATUS:=6176
112 PIAPB:=6178
113 ONLINE:=95
114 BUSY:=127
115 VRAGEN$=" vragen "
116 LEERLINGEN$=" leerlingen "
117 FOUTEN$=" fouten "
118 PRINT CHR$(10)
119 PRINT "FOUTEN STAAFDIAGRAM"
120 PRINT
121 //
122 EXEC: "voer het aantal in".VRAGEN$.NVRAGEN%
123 DIM FOUTEN%(1.NVRAGEN%)
124 DIM KARAKTER%(NVRAGEN%)
125 EXEC: "voer het aantal in".LEERLINGEN$.NLEERLING%
126 FOR VRAAG:=1 TO NVRAGEN%
127 PRINT "Voor vraag "VRAAG" ":
128 REPEAT
129 EXEC: "voer het aantal in".FOUTEN$.NFOUT%
130 UNTIL NFOUT%(<=NLEERLING%
131 FOUTEN%(0.VRAAG):=NFOUT%
132 FOUTEN%(1.VRAAG):=INT(100*NFOUT%/NLEERLING%+0.5)
133 ENDFOR
134 REPEAT

```



```

135 PRINT
136 PRINT "0 = absoluut afdrukken"
137 PRINT "1 = relatief afdrukken"
138 PRINT "2 = einde"
139 PRINT
140 PRINT " wat wilt u ? ":
141 GET N$
142 PRINT
143 KEUZE:=INT(VAL(N$))
144 IF KEUZE=0 OR KEUZE=1 THEN
145   EXEC: "diagram afdrukken"
146 ENDIF
147 UNTIL KEUZE=2
148 END.
149 //
150 //
151 PROC "diagram afdrukken"
152 PRINT "zet de printer aan a.u.b."
153 // wacht totdat de printer aan staat
154 REPEAT
155   A:=PEEK(STATUS)
156 UNTIL A=ONLINE
157 PRINT
158 EXEC: "init printer"
159 IF KEUZE=0 THEN
160   NREGELS%:=NLEERLING%
161 ELSE
162   NREGELS%:=34
163 ENDIF
164 FOR REGEL:=NREGELS% TO 1 STEP -1
165   EXEC: "construeer regel"
166   IF KEUZE=0 THEN
167     PRINT RIGHT$(" "+STR$(REGEL).2) " ":
168   ELSE
169     KARAKTER%(0):=240
170     EXEC: "print".KARAKTER%(0)
171     PRINT " ":
172   ENDIF
173   EXEC: "print regel"
174 ENDFOR
175 EXEC: "print onderlijn"
176 PRINT CHR$(10)
177 // short line
178 PRINT CHR$(27)"B"
179 // wacht totdat de printer uit staat
180 REPEAT
181   A:=PEEK(STATUS)
182 UNTIL A<>ONLINE AND A<>BUSY
183 ENDPROC
184 //
185 PROC "voer het aantal in".PR$.N%
186 REPEAT
187   PRINT
188   PRINT "Het aantal"PR$"a.u.b. ? ":
189   INPUT N$
190   N%:=INT(VAL(N$))
191 UNTIL VAL(N$)=N% AND N%>=0
192 ENDPROC
193 //
194 PROC "init printer"
195 IF NLEERLING%<30 THEN
196   // short line
197   PRINT CHR$(27)"B":
198 ELSE
199   // long line
200   PRINT CHR$(27)"A":
201 ENDIF
202 // 6 regels per inch
203 PRINT CHR$(27)"6":
204 // groot afdrukken
205 PRINT CHR$(31):
206 ENDPROC
207 //

```

=====

voor Junior met VDU-kaart
**** PACMAN ****

=====

De redactie hoopt binnenkort een serie publicaties uit te kunnen brengen van de inmiddels in de edities reeds bekende Phons Bloemen. Het betreft hier een artikel in 3 delen, buiten de tekst zo'n 3 x 20 K source omvattend. Het bekende soel kan alleen gedraaid worden als men de beschikking heeft over Ram-geheugen tot \$A000. Het gehele programma, inclusief de coding voor beeldscherm, de zoektafelen en het doolhof beslaat het geheugen van \$8000 tot en met \$9FFF. Ook de karaktergenerator EPROM van Elektuur (ESS 523) dient aanwezig te zijn.

Omdat er mensen zijn die het vele typewerk niet zien zitten, en omdat de laatste publicatie pas in april 1985 is te verwachten, heeft de redactie de routine op band beschikbaar gesteld.

Het staat in het bekende KIM en JUNIOR Hypertape formaat en kan met de monitor worden binnengehaald of met behulp van Micro-ADE.

De prijs van de cassette bedraagt:

Fl. 15,00

Voor de allerongeduldigsten stellen we de gehele source-listing, inclusief de inleiding en schema voor joystick ook beschikbaar via de papierware service. LET OP: het is precies hetzelfde als gepubliceerd zal worden. Het is dus niet nodig het te bestellen, maar het kan wel, als er teveel ongeduld is.

De prijs van de papieren bedraagt, inclusief verpakking en verzendkosten:

Fl. 45,00

Overschrijven op postgiro 841433 t.n.v. W.L. van Pelt te Krimpen a.d. IJssel, of op bankrekening 44.11.06.471 van de AMRO-bank te Krimpen a.d. IJssel (giro 3050). Een Eurocheque mag ook.

```

208 PROC "construeer regel"
209   FOR VRAAG:=1 TO NVRAGEN%
210     KARAKTER%(VRAAG):=128
211     IF KEUZE=0 THEN
212       IF FOUTEN%(0.VRAAG)=REGEL THEN
213         KARAKTER%(VRAAG):=255
214       ENDIF
215     ENDIF
216     IF KEUZE=1 THEN
217       IF FOUTEN%(1.VRAAG)=3*REGEL-2 THEN
218         KARAKTER%(VRAAG):=240
219       ENDIF
220       IF FOUTEN%(1.VRAAG)=3*REGEL-1 THEN
221         KARAKTER%(VRAAG):=252
222       ENDIF
223       IF FOUTEN%(1.VRAAG)=3*REGEL THEN
224         KARAKTER%(VRAAG):=255
225       ENDIF
226     ENDIF
227   ENDFOR
228 ENDPROC
229 //
230 PROC "print regel"
231   FOR VRAAG:=1 TO NVRAGEN%
232     EXEC: "print".KARAKTER%(VRAAG)
233   ENDFOR
234   PRINT
235 ENDPROC
236 //
237 PROC "print onderlijn"
238   KARAKTER%(0):=140
239   FOR VRAAG:=1 TO NVRAGEN%+3
240     EXEC: "print".KARAKTER%(0)
241   ENDFOR
242   // volgende regel normaal afdrukken
243   PRINT CHR$(10)CHR$(30) " ":
244   FOR VRAAG:=1 TO NVRAGEN%
245     PRINT RIGHT$(" "+STR$(VRAAG).2):
246   ENDFOR
247   PRINT
248 ENDPROC
249 //
250 PROC "print".CHARACTER%
251   REPEAT
252     A:=PEEK(STATUS)
253     UNTIL A<>BUSY
254     POKE PIAPB.CHARACTER%
255   ENDPROC
256 //

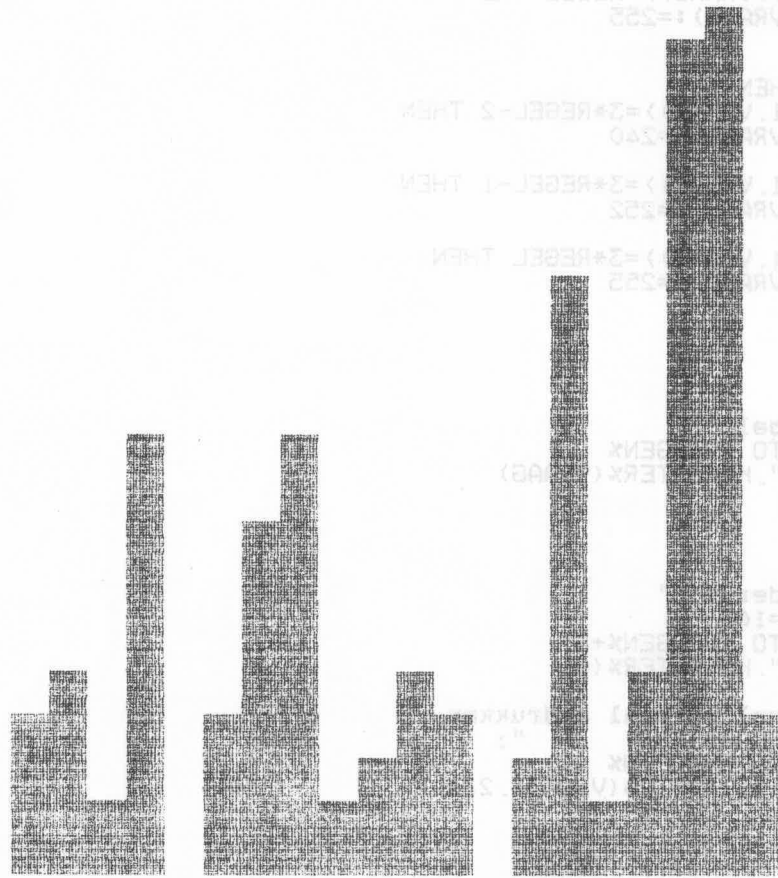
```

27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

0000
0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

PATCHES FOR SENIOR MONITOR

FATE 65XX ASSEMBLER V1.0 PAGE: 0001

```
0001 0000 .TIT 'PATCHES FOR SENIOR MONITOR'
0002 0000 .OPT GEN
0003 0000 .OPT PRI
0004 0000 .OPT SYM
0005 0000
0006 0000
0007 0000
0008 0000
0009 0000
0010 0000
0011 0000
0012 0000
0013 0000
```

```
*****
*
* PATCHES ON SENIOR MONITOR V4.6
*
*****
```

DEFINITIONS

```
0015 0000
0016 0000
0017 0000
0018 0000
0019 0000
0020 0000
0021 0000
0022 0000
0023 0000
0024 0000
0025 0000
0026 0000
0027 0000
```

CHARACTER DEFINITIONS

LF = \$0A : LINE FEED

SENIOR MONITOR LOCATIONS

TRFLG = \$0531 : TRACE FLAG
TRMCNT = \$0545 : HISTORY COUNTER
VFLG = \$058C : VERIFY FLAG
MAXLIN = \$058F : MAX. LINES/PAGE FOR PRINTER

FATE 65XX ASSEMBLER V1.0 PAGE: 0002

SENIOR MONITOR ROUTINES

NR. 1: ON ELEKTERMINAL MAXIMUM 16 LINES/SCREEN

*=\$E83E

```
LDA #18          : REPLACE BY 18. 14 LINES ON SCREEN
```

NR. 2: MAX 62 LINES/PAGE ON PRINTER.
NEW FORM FEED STRING

*=\$E26D

```

LDY #$06
NLS6 LDA INTAB.Y      : SET MAX. LINES/PAGE
STA MAXLIN.Y         : ON PRINTER AND FORM
DEY                  : FEED STRING

```

```
STA TRMCNT      : CLEAR HISTORY COUNTER
```

STA TRFLG 1.0141300: CLEAR TRACE FLAG

```
STA VFLG          : SET VERIFY FLAG
```

*=\$FFCA

```

NTAB      .BYT $3E      : MAX LINES/PAGE
          .BYT LF, LF, LF, LF, LF, $00

```

NR. 3: ON QUME DRIVES HEADS ARE ALWAYS LOADED. SO
THE HEAD LOAD TIME CAN BE 0 MSEC.
ON QUME DRIVES THE HEAD STEPPING TIME IS
BETTER THEN 15 MSEC.

*=\$FC64

```
LDA #0      : HEAD-TIME = 0 MSEC.
```

*=\$FDAF

LDA #3 : 15 MSEC. STEP TIME

.END

TIPS & TRICKS

OHIO DOS V3.X SPECIAL VOOR DE JUNIOR

Er zijn enkele kommando's die beginnen met de escape-toets gevolgd door een ander karakter. (b.v. ESC 1 = beeld wissen) De OHIO-basic negeert de escape-toets. Door nu te zeggen PRINT !(n) is dit opgelost. (n is de decimale waarde van het karakter dat na de escape-toets komt) Dus PRINT!(49) is gelijk aan print CHR\$(27);CHR\$(49).

Bij gebruik van de CTRL 'X' heb je grote kans dat het systeem hangt. Om dit te voorkomen zet dan vast in BEXEC de volgende instructies:

POKE 9593,234:POKE9594,234

Teruszetten in de oude stand gebeurt met:

POKE 9593,20:POKE9594,24

Na het invoeren van onderstaande instructies in de BEXEC kan de printer aan en uitgezet worden met CTRL 'G' (io wordt op ,09 gezet)

POKE 9610,201:POKE9611,7:POKE9618,9

Voor diegene die DOS V3.X hebben en de editor willen gebruiken, hij zit op track 15. Type in DISK!"io 15", run het en als het programma zich meldt met 'EDIT ENABLED' ,type dan het woord "HALLO" in, daarna CTRL 'H' gevolgd door CTRL 'P'. Nu is de editor geïntialiseerd.

Zo ook met de RUBOUT functie in DOS V3.X. Deze kun je gebruiken als men onderstaande instructies invoerd:

POKE1394,127:POKE1419,127

POKE2820,127:POKE1386,128

HIER VOLGEN NOG ENKELE ROUTINES IN DOS:

\$2D73 STRING OUT

JSR \$2D73 gevolgd door een tekststring wordt uitgeprint op het ingestelde device. De tekst-string moet eindigen op 00.

\$2761 UNLOAD HEAD

Gemakkelijk als het systeem hangt en de kop er nog op staat. (Sommige drives gaan dan ook niet open!)

\$267C 1 mSEC TELLER

\$2678 10 mSEC TELLER

BELANGRIJKE ADRESSEN:

\$2300 (8960D) HIMEM

Als achter in het RAM-gebied wat ruimte gereserveerd moet worden (b.v. in BASIC), bevat het HI-adres het laatste beschikbare RAM-gebied.

\$2322 (8994D) OUTPUT IO DEVICE

\$2DC4 (12716D) DIR TRACK

Hierin staat op welke track de directory te vinden is.

\$26A3 (9891D) TRACKACC

Track-to-track acces tijd in msec en decimaal. (standaard 28)

\$2EBC

Inhoud van \$31. Bij 2 Mc processors is dit \$62

Met dank aan W.v.Dinther

CHTAPE DE 6502 KENNER

PAGE 01

```

0010: 0200          CHTAPE ORG    $0200
0020:
0030:          *****
0040:          * CHECK TAPE *
0050:          *****
0060:
0070:          Phons Bloemen
0080:          Vinkelaan 200
0090:          5702 LX Helmond
0100:
0110:          This program reads a tape and gives
0120:          the ID (identificationnumber)
0130:          the SA (startaddress)
0140:          the EA (endaddress)
0150:          the CHK (checknumber)
0160:
0170:          So it serves everyones not knowing what
0180:          is on it.
0190:          To leave the program press (RST).
0200:          Parts of the program you'll find in TM.
0210:
0220:          ZERO PAGE
0230:          23 00 PRPNT *          $0023
0240:          FA 00 POINT *         $00FA
0250:          SUBROUTINES FROM TM
0260:          C2 0B RDBIT *          $0BC2 READS 1 BIT FROM TAPE
0270:          E8 0B BTWEEN *         $0BE8 DISPLAYS THE BETWEEN CHAR
0280:          F3 0B RDBYT *          $0BF3 READS 1 HEX BYTE = 2 ASCII CHAR FROM TAPE
0290:          36 0C READCH *         $0C36 READS AN ASCII CHAR FROM TAPE
0300:          4B 0C CHKSUM *         $0C4B COMPUTES CHECKSUM OF RECEIVED DATA
0310:          5D 0C CHARVU *         $0C5D OUTPUTS CHAR TO 7-SEGM DISPLAY
0320:          64 0C VU *             $0C64
0330:          SUBROUTINES FROM PM
0340:          E8 11 CALF *           $11E8 PRINT (CR) & (LF)
0350:          F3 11 PRSP *           $11F3 PRINT SPACE
0360:          8F 12 PRBYT *          $128F CONVERTS BYTE INTO TWO ASCII CHARS
0370:          34 13 PRCHA *          $1334 TRANSMIT ASCII CHAR TO PRINTER
0380:          BC 14 RESTTY *         $14BC
0390:          ALLOCATIONS IN PIA-RAM
0400:          58 1A CTRL *           $1A58
0410:          69 1A SY *             $1A69 SYN COUNTER
0420:          6B 1A CHAR *           $1A6B CHARACTER FROM TAPE
0430:          6E 1A CHKL *           $1A6E CHECKSUM LOW
0440:          6F 1A CHKH *           $1A6F CHECKSUM HIGH
0450:          70 1A SA *             $1A70 STARTADDRESS
0460:          78 1A GANG *           $1A78 TEMP OF PBD-BITS
0470:          79 1A ID *             $1A79
0480:          IN / OUT DEFINITIONS
0490:          81 1A PADD *           $1A81 DATA DIR.REG. OF PORT A
0500:          82 1A PBD *            $1A82 DATA REG. OF PORT B
0510:          83 1A PBDD *           $1A83 DATA DIR.REG. OF PORT B
0520:
0530:          MAIN PROGRAM
0540:
0550:          0200 A9 32 START LDAIM $32 INPUT RECORDER ON
0560:          0202 8D 82 1A STA PBD OUTPUT RECORDER OFF
0570:          0205 8D 78 1A STA GANG
0580:          0208 A9 7E LDAIM $7E PBO,PB7 INPUT
0590:          020A 8D 83 1A STA PBDD
0600:          020D A9 7F LDAIM $7F PA0..PA6 OUTPUT
0610:          020F 8D 81 1A STA PADD 7-SEGMENTS DISPLAY ON
0620:          0212 A9 00 LDAIM $00
0630:          0214 8D 6E 1A STA CHKL RESET CHK
0640:          0217 8D 6F 1A STA CHKH

```


CHTAPE DE 6502 KENNER

PAGE 02

```

0650: 021A A9 FF      SYNC   LDAIM $FF      RESET INCOMING
0660: 021C 8D 6B 1A    STA     CHAR      CHARACTER
0670: 021F 20 C2 0B    SYNCA   JSR      RDBIT
0680: 0222 6E 6B 1A    RDR      CHAR      SHIFT IN CHAR
0690: 0225 AD 6B 1A    LDA      CHAR      CHAR = SYNC ($16)?
0700: 0228 20 E8 0B    JSR      BTWEEN  DISPLAY BETWEENCHAR
0710: 022B C9 16      CMPIM  $16      SYNC?
0720: 022D D0 F0      BNE     SYNCA   GO ON UNTIL
0730: 022F A0 0A      LDYIM  $0A      10 SYNCs ON ROW
0740: 0231 8C 69 1A    STY     SY       SYNC COUNTER
0750: 0234 20 36 0C    TENSYN JSR      READCH  READ CHARACTER
0760: 0237 20 5D 0C    JSR      CHARVU  SYNCCHARACTER
0770: 023A C9 16      CMPIM  $16      STILL SYNC ?
0780: 023C D0 DC      BNE     SYNC    NO, AGAIN
0790: 023E CE 69 1A    DEC     SY       10 SYNCs ALREADY ?
0800: 0241 D0 F1      BNE     TENSYN
0810: 0243 20 36 0C    STAR   JSR      READCH  READ CHARACTER
0820: 0246 20 5D 0C    JSR      CHARVU  SYNCCHARACTER
0830: 0249 C9 2A      CMPIM  '*'      STAR ?
0840: 024B F0 06      BEQ     STARA   STAR
0850: 024D C9 16      CMPIM  $16      SYNC?
0860: 024F F0 F2      BEQ     STAR
0870: 0251 D0 AD      BNE     START   NOTHING? THEN AGAIN
0880: 0253 20 5D 0C    STARA  JSR      CHARVU  GOT IT
0890: 0256 20 F3 0B    JSR      RDBYT   READ BYTE (ID)
0900: 0259 8D 79 1A    STA     ID       SAVE IT
0910: 025C 20 F3 0B    JSR      RDBYT   READ BYTE (SAL)
0920: 025F 20 4B 0C    JSR      CHKSUM  ADJUST CHK
0930: 0262 85 FA      STAZ    POINT
0940: 0264 8D 70 1A    STA     SA
0950: 0267 20 F3 0B    JSR      RDBYT   READ BYTE (SAH)
0960: 026A 20 4B 0C    JSR      CHKSUM  ADJUST
0970: 026D 85 FB      STAZ    POINT   +01
0980: 026F 8D 71 1A    STA     SA       +01
0990: 0272 20 F3 0B    PNTUP  JSR      RDBYT   READ BYTE (DATA)
1000: 0275 30 29      BMI     BADTAP  ILLEGAL ?
1010: 0277 F0 0F      BEQ     CHECK   ENDCHARACTER / ?
1020: 0279 20 4B 0C    JSR      CHKSUM  ADJUST CHK
1030: 027C E6 FA      INCZ    POINT   POINT +1
1040: 027E D0 02      BNE     PNTJUM
1050: 0280 E6 FB      INCZ    POINT   +01
1060: 0282 20 64 0C    PNTJUM JSR      VU       GOT IT
1070: 0285 4C 72 02    JMP     PNTUP
1080: 0288 20 F3 0B    CHECK  JSR      RDBYT   READ BYTE (CHKL)
1090: 028B CD 6E 1A    CMP     CHKL     CHK CONTROL
1100: 028E D0 10      BNE     BADTAP
1110: 0290 20 F3 0B    JSR      RDBYT   READ BYTE (CHKH)
1120: 0293 CD 6F 1A    CMP     CHKH
1130: 0296 D0 08      BNE     BADTAP
1140: 0298 A9 00      LDAIM  $00      SET CTRL FLAG
1150: 029A 8D 58 1A    STA     CTRL
1160: 029D 4C A5 02    JMP     SCREEN
1170: 02A0 A9 01      BADTAP LDAIM  $01  SET CTRL FLAG
1180: 02A2 8D 58 1A    STA     CTRL
1190: 02A5 20 BC 14    SCREEN JSR      RESTTY  I/O FOR PM
1200: 02A8 20 E8 11    JSR      CRLF
1210: 02AB 20 1F 03    JSR      PRINT
1220: 02AE 49      =      'I
1230: 02AF 44      =      'D
1240: 02B0 3A      =      ':'
1250: 02B1 20      =      '
1260: 02B2 20      =      '
1270: 02B3 03      =      $03
1280: 02B4 AD 79 1A    LDA     ID       PRINT ID

```

CHTAPE DE 6502 KENNER

PAGE 03

1290:	02B7	20	8F	12	JSR	PRBYT	
1300:	02BA	AD	58	1A	LDA	CTRL	
1310:	02BD	FO	0E		BEQ	GOOD	
1320:	02BF	20	1F	03	JSR	PRINT	
1330:	02C2	20			=	,	
1340:	02C3	20			=	,	
1350:	02C4	42			=	'B	
1360:	02C5	41			=	'A	
1370:	02C6	44			=	'D	
1380:	02C7	20			=	,	
1390:	02C8	20			=	,	
1400:	02C9	03			=	\$03	
1410:	02CA	4C	00	02	JMP	START	
1420:	02CD	20	1F	03	JSR	PRINT	GOOD
1430:	02D0	20			=	,	
1440:	02D1	20			=	,	
1450:	02D2	4F			=	'O	
1460:	02D3	4B			=	'K	
1470:	02D4	20			=	,	
1480:	02D5	20			=	,	
1490:	02D6	20			=	,	
1500:	02D7	53			=	'S	
1510:	02D8	41			=	'A	
1520:	02D9	3A			=	'	
1530:	02DA	20			=	,	
1540:	02DB	20			=	,	
1550:	02DC	24			=	'\$	
1560:	02DD	03			=	\$03	
1570:	02DE	AD	71	1A	LDA	SA	+01 PRINT SA
1580:	02E1	20	8F	12	JSR	PRBYT	
1590:	02E4	AD	70	1A	LDA	SA	
1600:	02E7	20	8F	12	JSR	PRBYT	
1610:	02EA	20	1F	03	JSR	PRINT	
1620:	02ED	20			=	,	
1630:	02EE	20			=	,	
1640:	02EF	45			=	'E	
1650:	02F0	41			=	'A	
1660:	02F1	3A			=	'	
1670:	02F2	20			=	,	
1680:	02F3	20			=	,	
1690:	02F4	24			=	'\$	
1700:	02F5	03			=	\$03	
1710:	02F6	A5	FB		LDAZ	POINT	+01 PRINT EA.
1720:	02F8	20	8F	12	JSR	PRBYT	WAITING IN POINT
1730:	02FB	A5	FA		LDAZ	POINT	
1740:	02FD	20	8F	12	JSR	PRBYT	
1750:	0300	20	1F	03	JSR	PRINT	
1760:	0303	20			=	,	
1770:	0304	20			=	,	
1780:	0305	43			=	'C	
1790:	0306	48			=	'H	
1800:	0307	4B			=	'K	
1810:	0308	3A			=	'	
1820:	0309	20			=	,	
1830:	030A	20			=	,	
1840:	030B	24			=	'\$	
1850:	030C	03			=	\$03	
1860:	030D	AD	6F	1A	LDA	CHKH	PRINT CHK
1870:	0310	20	8F	12	JSR	PRBYT	
1880:	0313	AD	6E	1A	LDA	CHKL	
1890:	0316	20	8F	12	JSR	PRBYT	
1900:	0319	20	F3	11	JSR	PRSP	
1910:	031C	4C	00	02	JMP	START	
1920:							

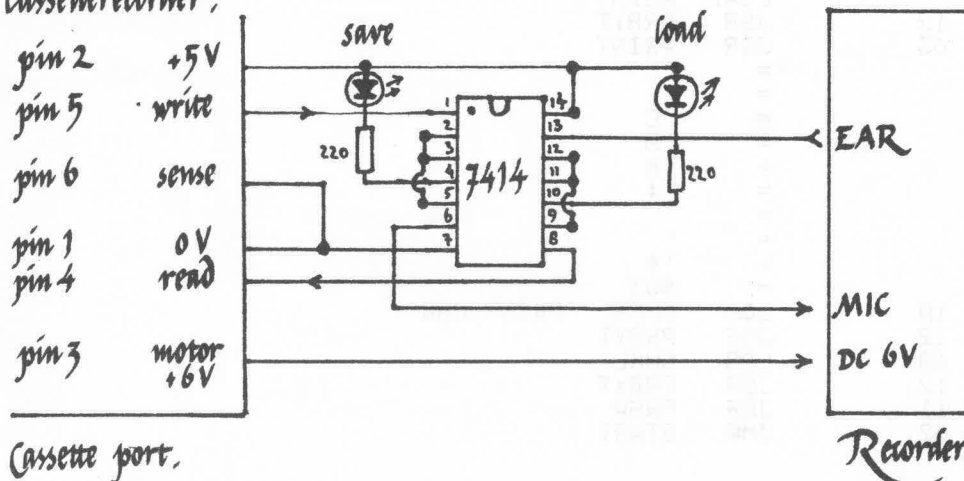
```

1930:                               END PROGRAM
1940:
1950:                               PRINT
1960:                               See Junior Paperware 1 or 6502-K nr 25
1970: 031F 68                       PRINT PLA
1980: 0320 85 23                     STAZ PRPNT
1990: 0322 68                       PLA
2000: 0323 85 24                     STAZ PRPNT +01
2010: 0325 E6 23                     PRTA INCZ PRPNT
2020: 0327 D0 02                     BNE PRTB
2030: 0329 E6 24                     INCZ PRPNT +01
2040: 032B A0 00                     PRTB LDYIM $00
2050: 032D B1 23                     LDAIY PRPNT
2060: 032F C9 03                     CMPIM $03
2070: 0331 F0 06                     BEQ PRTC
2080: 0333 20 34 13                 JSR PRCHA
2090: 0336 4C 25 03                 JMP PRTA
2100: 0339 A5 24                     PRTC LDAZ PRPNT +01
2110: 033B 48                       PHA
2120: 033C A5 23                     LDAZ PRPNT
2130: 033E 48                       PHA
2140: 033F 60                       RTS
    
```

Cassette - interface : simpel.

Na in een vorig nummer getoond te hebben met welke interface-schakeling ik mijn normale audio-cassetterecorder op de CBM 64 heb aangesloten, ben ik nog wat aan het experimenteren geslagen. Zelf heb ik 'Realistic' CTR 39 recorders die op 6 Volt voedingsspanning werken. Dat is wat de 64 levert en dus de moeite van het uitproberen waard.

Met één TTL-IC blijkt het mogelijk betrouwbaar te werken met een normale cassette recorder:



Cassette port.

Recorder.

Voor de 64 lijkt het nu net alsof er altijd een toets op de recorder is ingedrukt, dus zelf op de toetsen letten bij LOAD en SAVE!

Fer Weber

0 LIST

SCR # 0

0 -->

1 DEZE FORTH WOORDEN MAKEN HET MOGELIJK OM EEN HEXDUMP
2 EN EEN ASCIIDUMP AF TE DRUKKEN

3

4 HEXDUMP (ADR1 ADR2 --)

5 DRUKT DE HEXDUMP TUSSEN ADRES ADR1 EN ADR2
6 AF.

7 ASCDUMP (ADR1 ADR2 --)

8 DRUKT DE ASCIIDUMP TUSSEN ADRES ADR1 EN ADR2
9 AF.

10

11 GERT KLEIN

12 DIEDENWEG 119

13 6706 CM WAGENINGEN

14 02-06-1984

15

OK

1 LIST

SCR # 1

0 (HEXDUMP 1

GKL JUNI 84)

1 0 VARIABLE POINT

2 0 VARIABLE ENDAD

3 HEX

4 : U. 0 D. : (PRINT AN UNSIGNED NUMBER)

5 : FETCHBYTE POINT @ C@ : (FETCH BYTE FROM POINT)

6 : .0 0 DO 30 EMIT LOOP : (PRINT N NULLSZEROES)

7 : .POINTER POINT @ (PRINT POINTER WITH LEADING ZEROES)

8 DUP 10 < OVER FFFF > AND IF 3 .0 ENDIF

9 DUP 100 < OVER OF > AND IF 2 .0 ENDIF

10 DUP 1000 < OVER FF > AND IF 1 .0 ENDIF

11 U. :

12 (PRINT TOP ROW 0 TO F)

13 : .ROW 5 SPACES 10 0 DO I . SPACE LOOP CR :

14 : ?ASCII DUP 7F < OVER 20 > AND : (VALID ASCII ?)

15 DECIMAL -->

OK

2 LIST

SCR # 2

0 (HEXDUMP 2

GKL JUNI 84)

1 : HEXDUMP ENDAD ! POINT ! CR CR .ROW (SAVE POINTERS)

2 BEGIN CR .POINTER 16 0 (PRINT 16 BYTES)

3 DO FETCHBYTE DUP 16 < (LEADING ZERO ?)

4 IF (YES)

5 48 EMIT (PRINT ZERO)

6 ENDIF

7 . 1 POINT +! (INCREMENT POINT)

8 POINT @ ENDAD @ = (DONE ?)

9 IF (YES)

10 QUIT (TERMINATE PROGRAM)

11 ENDIF

12 LOOP (NEXT BYTE)

13 AGAIN : (ALWAYS)

14

15 -->

OK

3 LIST

SCR # 3

0 (ASCII DUMP

GKL JUNI 84)

1 : ASCDUMP ENDAD ! POINT ! CR CR .ROW (SAVE POINTERS)

2 BEGIN CR .POINTER 16 0 (16 CHARACTERS)

3 DO FETCHBYTE ?ASCII (PRINTABLE ASCII ?)

4 IF (YES)

5 SPACE EMIT SPACE (OUTPUT IT)

6 ELSE (NOT PRINTABLE)

7 DROP 3 SPACES (OUTPUT BLANKS)

8 ENDIF

9 . 1 POINT +! (POINTER + 1)

10 POINT @ ENDAD @ = (DONE ?)

11 IF (YES)

12 QUIT

13 ENDIF

14 LOOP

15 AGAIN (ALWAYS) : HEX ;S

OK

SYM-1 Basic

Tokenized Microsoft Basic Keywords and addresses

W.L. van Pelt

Analogous to the publication for Commodore-64 by A. Mueller,

DE 6502 KENNER, December 1983, pages 5-8.

COMMANDS		
KEYWORDS CORRESPONDING TO C089		
ADDRESSES CORRESPONDING TO C003		
THE ADDRESSES OF ROUTINES FOR COMMANDS ARE THE ADDRESSES		
MINUS 1, BECAUSE THE ROUTINES ARE INVOKED THROUGH RTS.		

KEYWORD	TOKEN	ADDR-1			
END	80	C623	WAIT	92	D5E2
FOR	81	C534	LOAD	93	C6B6
NEXT	82	CAD7	SAVE	94	C675
DATA	83	C781	VERIFY	-	-
INPUT#	-	-	DEF	95	D16B
INPUT	84	C9B8	POKE	96	D5D9
DIM	85	CE54	PRINT#	-	-
READ	86	C9E4	PRINT	97	C8BD
LET	87	C82E	CONT	98	C64A
GOTO	88	C72E	LIST	99	C4AC
RUN	89	C706	CLR	9A	C471
IF	8A	C7B1	CMD	-	-
RESTORE	8B	C609	SYS	-	-
GOSUB	8C	C711	OPEN	-	-
RETURN	8D	C75B	CLOSE	-	-
REM	8E	C7C4	GET	9B	D001
STOP	8F	C621	NEW	9C	C455
ON	90	C7D4			
NULL	91	C664			

MISCELLANEOUS KEYWORDS	
KEYWORDS CORRESPONDING TO C0F9	

KEYWORD	TOKEN
TAB(9D
TO	9E
FN	9F
SPC(A0
THEN	A1
NOT	A2
STEP	A3

SYM-1 Basic

Tokenized Microsoft Basic keywords and addresses

-----		0000	0000
		0001	0001
	DYADIC OPERATORS	0002	0002
		0003	0003
	KEYWORDS CORRESPONDING TO C110	0004	0004
		0005	0005
	PRIORITIES AND ADDRESSES CORRESPONDING TO C06B	0006	0006
		0007	0007
	THE ADDRESSES OF ROUTINES FOR DYADIC OPERATORS ARE THE	0008	0008
	ADDRESS MINUS 1, BECAUSE THE ROUTINES ARE INVOKED	0009	0009
	THROUGH A RTS INSTRUCTION.	0010	0010
		0011	0011
		0012	0012
		0013	0013
		0014	0014
		0015	0015
		0016	0016
		0017	0017
		0018	0018
		0019	0019
		0020	0020
		0021	0021
		0022	0022
		0023	0023
		0024	0024
		0025	0025
		0026	0026
		0027	0027
		0028	0028
		0029	0029
		0030	0030
		0031	0031
		0032	0032
		0033	0033
		0034	0034
		0035	0035
		0036	0036
		0037	0037
		0038	0038
		0039	0039
		0040	0040
		0041	0041
		0042	0042
		0043	0043
		0044	0044
		0045	0045
		0046	0046
		0047	0047
		0048	0048
		0049	0049
		0050	0050
		0051	0051
		0052	0052
		0053	0053
		0054	0054
		0055	0055
		0056	0056
		0057	0057
		0058	0058
		0059	0059
		0060	0060
		0061	0061
		0062	0062
		0063	0063
		0064	0064
		0065	0065
		0066	0066
		0067	0067
		0068	0068
		0069	0069
		0070	0070
		0071	0071
		0072	0072
		0073	0073
		0074	0074
		0075	0075
		0076	0076
		0077	0077
		0078	0078
		0079	0079
		0080	0080
		0081	0081
		0082	0082
		0083	0083
		0084	0084
		0085	0085
		0086	0086
		0087	0087
		0088	0088
		0089	0089
		0090	0090
		0091	0091
		0092	0092
		0093	0093
		0094	0094
		0095	0095
		0096	0096
		0097	0097
		0098	0098
		0099	0099
		0100	0100

KEYWORD	TOKEN	ADDR-1	PRTY	

+	A4	D61F	79	addition
-	A5	D608	79	subtraction
*	A6	D7E0	7B	multiplication
/	A7	D8C7	7B	division
^	A8	DCFC	7F	exponentiation
AND	A9	CD27	50	logical AND
OR	AA	CD24	46	logical OR
monadic "--"	AB	DD35	7D	negation
monadic NOT	AC	CC73	5A	logical NOT
)=(AD	CD54	64	comparison

-----		0000	0000
		0001	0001
	FUNCTIONS	0002	0002
		0003	0003
	FUNCTIONS CORRESPONDING TO C11D	0004	0004
		0005	0005
	ADDRESSES CORRESPONDING TO C03D	0006	0006
		0007	0007
		0008	0008
		0009	0009
		0010	0010
		0011	0011
		0012	0012
		0013	0013
		0014	0014
		0015	0015
		0016	0016
		0017	0017
		0018	0018
		0019	0019
		0020	0020
		0021	0021
		0022	0022
		0023	0023
		0024	0024
		0025	0025
		0026	0026
		0027	0027
		0028	0028
		0029	0029
		0030	0030
		0031	0031
		0032	0032
		0033	0033
		0034	0034
		0035	0035
		0036	0036
		0037	0037
		0038	0038
		0039	0039
		0040	0040
		0041	0041
		0042	0042
		0043	0043
		0044	0044
		0045	0045
		0046	0046
		0047	0047
		0048	0048
		0049	0049
		0050	0050
		0051	0051
		0052	0052
		0053	0053
		0054	0054
		0055	0055
		0056	0056
		0057	0057
		0058	0058
		0059	0059
		0060	0060
		0061	0061
		0062	0062
		0063	0063
		0064	0064
		0065	0065
		0066	0066
		0067	0067
		0068	0068
		0069	0069
		0070	0070
		0071	0071
		0072	0072
		0073	0073
		0074	0074
		0075	0075
		0076	0076
		0077	0077
		0078	0078
		0079	0079
		0080	0080
		0081	0081
		0082	0082
		0083	0083
		0084	0084
		0085	0085
		0086	0086
		0087	0087
		0088	0088
		0089	0089
		0090	0090
		0091	0091
		0092	0092
		0093	0093
		0094	0094
		0095	0095
		0096	0096
		0097	0097
		0098	0098
		0099	0099
		0100	0100

KEYWORD	TOKEN	ADDR	

SGN	AE	D9EF	TAN
INT	AF	DA82	ATN
ABS	B0	DA0E	PEEK
USR	B1	000A	LEN
FRE	B2	D138	STR\$
POS	B3	D159	VAL
SQR	B4	DCF3	ASC
RND	B5	DE14	CHR\$
LOG	B6	D7A0	LEFT\$
EXP	B7	DD6F	RIGHT\$
COS	B8	00C3	MID\$
SIN	B9	00C3	

Note: 1) Bit 7 in the last character of each keyword is set to determine the end of a keyword

2) Keywords are tokenized by adding x'80' to their relative (hex) position in the table.

BASIC-SAVE- AND LOAD

PROTON 650X ASSEMBLER V4.4 PAGE: 0001

```

0001 0000 .TIT 'BASIC SAVE- AND LOAD
0002 0000 .OPT GEN
0003 0000 .OPT SYM
0004 0000
0005 0000 *****
0006 0000
0007 0000 UITBREIDING OP DE SYM-1 BASIC VOOR KIM-1 EN JUNIOR
0008 0000
0009 0000 *** SAVE- EN LOAD ROUTINES NAAR FLOPPY-DISK ***
0010 0000
0011 0000 *****
0012 0000
0013 0000 AUTEUR: F.J.M. SMEEHUIJZEN
0014 0000 LIPPEDAL 19
0015 0000 2904 CL CAPELLE AAN DEN IJSSEL
0016 0000 TEL: 010-512507
0017 0000
0018 0000 DE NU VOLGENDE ROUTINES ZIJN GESCHREVEN VOOR EEN SYSTEEM
0019 0000 DAT GEBRUIK MAAKT VAN DE PROTON-MONITOR.
0020 0000 VANDAAR EEN KORTE UITLEG VAN DE GEBRUIKTE SUBROUTINES.
0021 0000
0022 0000 - MET DE ROUTINE 'OUTFLO' WORDT HET OUTPUT-DEVICE BEPAALD
0023 0000 WAARNAAR GESCHREVEN MOET WORDEN.
0024 0000 - MET DE ROUTINE 'INFLOP' WORDT EEN OUTPUT-DEVICE BEPAALD
0025 0000 WAARVAN GELEZEN MOET WORDEN.
0026 0000 - MET DE ROUTINE 'FLOPO' WORDT GEVRAAGD NAAR DE FILENAME
0027 0000 VAN HET WEG TE SCHRIJVEN BESTAND.
0028 0000 - MET DE ROUTINE 'FLOPI' WORDT GEVRAAGD NAAR DE FILENAME
0029 0000 VAN HET IN TE LEZEN BESTAND.
0030 0000 - MET DE ROUTINE 'CLOSED' WORDT HET OUTPUT-DEVICE AF-
0031 0000 GESLOTEN.
0032 0000 - MET DE ROUTINE 'OUTALL' WORDT DE ACCU-INHOUD NAAR HET
0033 0000 OUTPUT-DEVICE GESCHREVEN.
0034 0000 - MET DE ROUTINE 'INALL' WORDT DE INHOUD VAN HET VAN TE
0035 0000 VOREN OPGEGEVEN INPUT-DEVICE OPGEHAALD.
0036 0000 - MET DE ROUTINE 'CLOSEI' WORDT HET INPUT-DEVICE
0037 0000 AFGESLOTEN.
0038 0000 - MET DE ROUTINE 'HEAD' WORDT DE LEES/SCHRIJFKOP VAN DE
0039 0000 FLOPPY VAN DE SCHIJF GELICHT EN DE MOTOR GESTOPT.
0040 0000
0041 0000 DE SAVE-ROUTINE WORDT OPGESTART DOOR HET INTOETSEN VAN HET
0042 0000 KOMMANDO - #SAVE -
0043 0000 WAARNA VERVOLGENS NAAR ADRES $99E8 WORDT GESPRONGEN.
0044 0000
0045 0000 DE LOAD-ROUTINE WORDT OPGESTART DOOR HET INTOETSEN VAN HET
0046 0000 KOMMANDO - #LOAD -
0047 0000 WAARNA VERVOLGENS NAAR ADRES $9A1A WORDT GESPRONGEN.
0048 0000
0049 0000 ZIE VOOR BEIDE KOMMANDO'S DE BESCHRIJVING VAN HET TOEVOEGE
0050 0000 VAN NIEUWE BASIC KOMMANDO'S IN K.K. 29.
0051 0000
0052 0000
0053 0000 *** ZERO PAGE DEFINITIONS ***
0054 0000
0055 0000 .EX1
0056 0000
0057 0000 **=$0060
0058 0060
0059 0060 TEMPA **=$+2 ; TEMPORARY ADDRESS POINTER 1
0060 0062 TEMPB **=$+25 ; TEMPORARY ADDRESS POINTER 2
0061 007B BOBL **=$+2 ; BEGIN OF BASIC SOURCE LOW
0062 007D EOBL **=$+0 ; END OF BASIC SOURCE LOW
0063 007D BODL **=$+4 ; BEGIN OF BASIC DATA LOW
0064 0081 EODL **=$+2 ; END OF BASIC DATA LOW
0065 0083 BOSL **=$+4 ; BEGIN OF BASIC STRING AREA LOW
0066 0087 EOSL **=$+1 ; END OF BASIC STRING AREA LOW
0067 0088
0068 0088 **=$C27E
0069 C27E
0070 C27E WARM ; BASIC WARM START
0071 C27E
0072 C27E **=$99E8
0073 99E8
0074 99E8
0075 99E8

```

BASIC SAVE- AND LOAD

PROTON 650X ASSEMBLER V4.4 PAGE: 0002

```

0076 99E8 20ED9A SAVEP JSR OUTFLO ; OPEN OUTPUT DEVICE
0077 99EB 20F09A JSR FLOPO ; ASK FOR FILENAME
0078 99EE 20359A JSR PRGOUT ; FILL POINTERS BASIC SOURCE AREA
0079 99F1 20BE9A JSR DATOUT ; SAVE PROGRAM
0080 99F4 20679A JSR NULL ; FILL POINTERS ZERO PAGE AREA
0081 99F7 20BE9A JSR DATOUT ; SAVE ZERO PAGE
0082 99FA 20F39A JSR CLOSED ; CLOSE FILE
0083 99FD 20FF9A JSR HEAD ; DESELECT FLOPPY
0084 9A00 60 RTS
0085 9A01 20ED9A SAVED JSR OUTFLO ; OPEN OUTPUT DEVICE
0086 9A04 20F09A JSR FLOPO ; ASK FOR FILENAME
0087 9A07 20B49A JSR BDATA ; FILL POINTERS BASIC DATA AREA
0088 9A0A 20BE9A JSR DATOUT ; SAVE DATA AREA
0089 9A0D 20A19A JSR BSTRIN ; FILL POINTERS BASIC STRING AREA
0090 9A10 20BE9A JSR DATOUT ; SAVE STRING AREA
0091 9A13 20F39A JSR CLOSED ; CLOSE FILE
0092 9A16 20FF9A JSR HEAD ; DESELECT FLOPPY
0093 9A19 60 RTS
0094 9A1A ;
0095 9A1A ; *** LOAD-ROUTINE VOOR DE SYM-1 BASIC ***
0096 9A1A ;
0097 9A1A 20029B LOADP JSR INFLOP ; OPEN INPUT DEVICE
0098 9A1D 20059B JSR FLOPI ; ASK FOR FILENAME
0099 9A20 20529A JSR PRGIN ; LOAD START ADDRESS BASIC SOURCE
0100 9A23 20CB9A JSR DATAIN ; LOAD DATA
0101 9A26 20529A JSR PRGIN ; LOAD START ADDRESS ZERO PAGE
0102 9A29 20CB9A JSR DATAIN ; LOAD DATA
0103 9A2C 20FC9A JSR CLOSEI ; CLOSE INPUT DEVICE
0104 9A2F 20FF9A JSR HEAD ; DESELECT FLOPPY
0105 9A32 4C7EC2 JMP WARM
0106 9A35 ;
0107 9A35 ; *** SUBROUTINE TO WRITE START- AND END ***
0108 9A35 ; *** ADDRESS OF BASIC SOURCE TO FLOPPY ***
0109 9A35 ;
0110 9A35 A57B PRGOUT LDA BOBL ; BEGIN ADDRESS LOW
0111 9A37 8560 STA TEMPA ; COPY ADDRESS
0112 9A39 20F69A JSR OUTALL ; WRITE TO FLOPPY
0113 9A3C A57C LDA BOBL+1 ; BEGIN ADDRESS HIGH
0114 9A3E 8561 STA TEMPA+1 ; COPY ADDRESS
0115 9A40 20F69A JSR OUTALL ; WRITE TO FLOPPY
0116 9A43 A57D LDA EOBL ; END ADDRESS LOW
0117 9A45 20F69A JSR OUTALL ; WRITE TO FLOPPY
0118 9A48 8562 STA TEMPB ; COPY ADDRESS
0119 9A4A A57E LDA EOBL+1 ; END ADDRESS HIGH
0120 9A4C 20F69A JSR OUTALL ; WRITE TO FLOPPY
0121 9A4F 8563 STA TEMPB+1 ; COPY ADDRESS
0122 9A51 60 RTS
0123 9A52 ;
0124 9A52 ; *** SUBROUTINE TO READ START- AND END ***
0125 9A52 ; *** ADDRESS OF BASIC SOURCE FROM FLOPPY ***
0126 9A52 ;
0127 9A52 20F99A PRGIN JSR INALL ; READ FROM FLOPPY
0128 9A55 8560 STA TEMPA ; STORE START ADDRESS LOW
0129 9A57 20F99A JSR INALL ; READ FROM FLOPPY
0130 9A5A 8561 STA TEMPA+1 ; STORE START ADDRESS HIGH
0131 9A5C 20F99A JSR INALL ; READ FROM FLOPPY
0132 9A5F 8562 STA TEMPB ; STORE END ADDRESS LOW
0133 9A61 20F99A JSR INALL ; READ FROM FLOPPY
0134 9A64 8563 STA TEMPB+1 ; STORE END ADDRESS HIGH
0135 9A66 60 RTS
0136 9A67 ;
0137 9A67 ; *** SUBROUTINE TO SAVE ZERO-PAGE POINTERS ***
0138 9A67 ;
0139 9A67 A965 NULL LDA #65 ; START OF ZERO-PAGE POINTERS
0140 9A69 8560 STA TEMPA ;
0141 9A6B 20F69A JSR OUTALL ; WRITE TO FLOPPY
0142 9A6E A900 LDA #00 ;
0143 9A70 8561 STA TEMPA+1 ;
0144 9A72 20F69A JSR OUTALL ; WRITE TO FLOPPY
0145 9A75 A9EA LDA #EA ; END OF ZERO-PAGE POINTERS
0146 9A77 8562 STA TEMPB ;
0147 9A79 20F69A JSR OUTALL ; WRITE TO FLOPPY
0148 9A7C A900 LDA #00 ;
0149 9A7E 8563 STA TEMPB+1 ;
0150 9A80 20F69A JSR OUTALL ; WRITE TO FLOPPY
0151 9A83 60 RTS
0152 9A84 ;
0153 9A84 ; *** WRITE START- AND ENDADDRESS OF ***

```

```

0154 9A84      ; *** BASIC DATA AREA TO FLOPPY ***
0155 9A84      ;
0156 9A84 A57D BDATA LDA BDDL      ; BEGIN OF BASIC DATA AREA LOW
0157 9A86 8560 STA TEMPB
0158 9A88 20F69A JSR OUTALL      ; WRITE TO FLOPPY
0159 9A8B A57E LDA BDDL+1      ; BEGIN OF BASIC DATA AREA HIGH
0160 9A8D 8561 STA TEMPB+1
0161 9A8F 20F69A JSR OUTALL      ; WRITE TO FLOPPY
0162 9A92 A581 LDA EDDL      ; END OF BASIC DATA AREA LOW
0163 9A94 8562 STA TEMPB
0164 9A96 20F69A JSR OUTALL      ; WRITE TO FLOPPY
0165 9A99 A582 LDA EDDL+1      ; END OF BASIC DATA AREA HIGH
0166 9A9B 8563 STA TEMPB+1
0167 9A9D 20F69A JSR OUTALL      ; WRITE TO FLOPPY
0168 9AA0 60 RTS
0169 9AA1      ;
0170 9AA1      ; *** WRITE START- AND ENDADDRESS OF ***
0171 9AA1      ; *** BASIC STRING AREA TO FLOPPY ***
0172 9AA1      ;
0173 9AA1 A583 BSTRIN LDA BBSL      ; BEGIN OF BASIC STRING AREA LOW
0174 9AA3 8560 STA TEMPB
0175 9AA5 20F69A JSR OUTALL      ; WRITE TO FLOPPY
0176 9AA8 A584 LDA BBSL+1      ; BEGIN OF BASIC STRING AREA HIGH
0177 9AAA 8561 STA TEMPB+1
0178 9AAC 20F69A JSR OUTALL      ; WRITE TO FLOPPY
0179 9AAF A587 LDA EOSL      ; END OF BASIC STRING AREA LOW
0180 9AB1 8562 STA TEMPB
0181 9AB3 20F69A JSR OUTALL      ; WRITE TO FLOPPY
0182 9AB6 A588 LDA EOSL+1      ; END OF BASIC STRING AREA HIGH
0183 9AB8 8563 STA TEMPB+1
0184 9ABA 20F69A JSR OUTALL      ; WRITE TO FLOPPY
0185 9ABD 60 RTS
0186 9ABE      ;
0187 9ABE      ; *** OUTPUT DATA TO FLOPPY ***
0188 9ABE      ;
0189 9ABE A000 DATOUT LDY #00
0190 9AC0 B160 LDA (TEMPB),Y
0191 9AC2 20F69A JSR OUTALL      ; WRITE TO FLOPPY
0192 9AC5 20D89A JSR ADRES      ; CALCULATE END OF DATA
0193 9AC8 B0F4 BCS DATOUT
0194 9ACA 60 RTS
0195 9ACB      ;
0196 9ACB      ; *** INPUT DATA FROM FLOPPY ***
0197 9ACB      ;
0198 9ACB A000 DATAIN LDY #00
0199 9ACD 20F99A JSR INALL      ; READ FROM FLOPPY
0200 9AD0 9160 STA (TEMPB),Y
0201 9AD2 20D89A JSR ADRES      ; CALCULATE END OF DATA
0202 9AD5 B0F4 BCS DATAIN
0203 9AD7 60 RTS
0204 9AD8      ;
0205 9AD8      ; *** INCREMENT DATA POINTER ***
0206 9AD8      ;
0207 9AD8 18 ADRES CLC
0208 9AD9 A901 LDA #01      ; ADD 1 TO TEMPORARY POINTER
0209 9ADB 6560 ADC TEMPB
0210 9ADD 9002 BCC ADR1
0211 9ADF E661 INC TEMPB+1
0212 9AE1 8560 ADR1 STA TEMPB
0213 9AE3 38 SEC      ; COMPARE BEGIN- AND END POINTER
0214 9AE4 A562 LDA TEMPB
0215 9AE6 E560 SBC TEMPB
0216 9AE8 A563 LDA TEMPB+1
0217 9AEA E561 SBC TEMPB+1
0218 9AEC 60 RTS
0219 9AED      ;
0220 9AED      ; *** SENIOR MONITOR ROUTINES ***
0221 9AED      ;
0222 9AED 6C38E0 OUTFLO JMP (#E038)
0223 9AF0 6C3CE0 FLOPO JMP (#E03C)
0224 9AF3 6C0CE0 CLOSED JMP (#E00C)
0225 9AF6 6C08E0 OUTALL JMP (#E008)
0226 9AF9 6C06E0 INALL JMP (#E006)
0227 9AFC 6C0AE0 CLOSEI JMP (#E00A)
0228 9AFF 6C28E0 HEAD JMP (#E028)
0229 9B02 6C36E0 INFLOP JMP (#E036)
0230 9B05 6C3AE0 FLOPI JMP (#E03A)
0231 9B08      ;

```


ASS L

```

0010 : WALLBREAKER FOR JUNIOR WITH OHIO-DOS
0020 : =====
0030 :
0040 : THIS PROGRAM IS A LITTLE DERIVATION OF THE
0050 : VIDEOGAME 'BREAKOUT'.
0060 : BREAK THE WALL. FOR EACH STONE YOU'LL GET
0070 : POINTS.
0080 :
0090 : STARTADDRESS=$3A7E. AS SOON AS THE CANON
0100 : GOES ON AND ON. YOU MAY SHOOT WITH (BRK).
0110 :
0120 : THE ROUTINE FOR JUNIOR WAS PUBLISHED BEFORE
0130 : IN THE 6502 KENNER. EDITION NR. 22. AUGUST
0140 : 1982. BY W.J. KUITENS.
0150 : NOW MODIFIED FOR JUNIOR WITH OHIO-DOS BY:
0160 :
0170 : HENK MOCKING
0180 : 't ZAND 42
0190 : 3451 GS VLEUTEN
0200 :
0210 PRCHA .DE $FEA3 : PRINT CHAR FROM ACCU
0220 CNTD .DE $FAF7 : COUNT 1024T NO IRQ
0230 RDFLAG .DE $FAD5 : FLAG-REGISTER
0240 LFEED .DE $2D6F : PRINT LINEFEED
0250 CRLF .DE $2D6A : PRINT A (CR/LF)
0260 BRKL .DE $FA7C : BREAK VECTOR
0270 BRKH .DE $FA7D :
0280 PRBYT .DE $2D92 : PRINT BYTE FROM ACCU
0290 POINTL .DE $00FA : ADDRESS POINTER
0300 POINTH .DE $00FB :
0310 PNT1 .DE $0000 : 'STAND' REGISTER
0320 PNT2 .DE $0001 :
0330 WINST .DE $0002 : 'WINST' REGISTER
0340 DWNEXT .DE $0003 : EXTRA (LF) REGISTER
0350 TEMPA .DE $FA60 : TEMPX IN PRCHA
0360 :
0370 : .BA $3A7E
0380 :
0390 WALL LDA #$0C : CLEAR SCREEN
0400 JSR PRCHA
0410 LDA #$83 : 132 ms TILL SCREEN IS CLEAR
0420 STA CNTD
0430 DELA BIT RDFLAG : WHEN DELA IS OVER. CONTINUE
0440 BPL DELA
0450 LDX #$40 : FROM HERE ON WE ARE
0460 STX *POINTH : GOING TO FILL LINES
0470 VUL LDX #0 : WITH '===' AND
0480 STX *POINTL : ALSO STORE IT IN
0490 STX *PNT1 : MEMORY. FROM $4001...$403F
0500 STX *PNT2 : $4101...$413F/$4201...$423F
0510 VL LDA #$3D : $4301...$433F/$4401...$443F
0520 LDY #0 : ALSO SCORE TOTAL IS
0530 INC *POINTL : SET TO 0000.
0540 STA (POINTL).Y
0550 JSR PRCHA
0560 LDA *POINTL
0570 CMP #$3F : 3F HEX = 63 DEC. ELEKTERMINAL
0580 BNE VL : MAX. 64 CHAR. ON A LINE. ON LAST
0590 JSR CRLF : POS. (64) NO "="
0600 INC *POINTH
0610 LDA *POINTH
0620 CMP #$45 : WALL BUILD ?
0630 BNE VUL
0640 :
0650 :
0660 : THE WALL SHOULD BE BUILD NOW. AND STORED IN MEMORY
0670 :
0680 CURDWN LDX #$0A : IF WALL BUILD. GO TO LINE 16
0690 LF JSR LFEED
0700 DEX
0710 BNE LF : ARE WE ON LINE 16 ?

```

3ABD-	A9 E7	0720	LDA #E7	
3ABF-	AD 7C FA	0730	STA BRKL	
3AC2-	A9 3A	0740	LDA #3A	
3AC4-	8D 7D FA	0750	STA BRKH	
		0760 :		
3AC7-	A2 01	0770	LDX #1	: CANON MOVES ACROSS LINE 16
3AC9-	A9 0D	0780	LDA #0D	: GO TO START OF LINE 16
3ACB-	20 A3 FE	0790	JSR PRCHA	
3ACE-	A9 55	0800	LDA #55	: 'U' IS CANON
3AD0-	20 A3 FE	0810	JSR PRCHA	
3AD3-	A9 08	0820	LDA #8	: REMOVE 'U' FROM SCREEN
3AD5-	20 A3 FE	0830	JSR PRCHA	
3AD8-	A9 20	0840	LDA #20	
3ADA-	20 A3 FE	0850	JSR PRCHA	
3ADD-	E8	0860	INX :	NEXT POSITION OF CANON AND REG.X
3ADE-	E0 40	0870	CPX #40	: DID WE REACH END OF LINE OR FIRE
3AE0-	F0 E5	0880	BEQ LOOP1	: BUTTON (BRK) HIT ?
3AE2-	B0 28	0890	BCS SHOOT	: IF SO. ***FIRE!***
3AE4-	4C CE 3A	0900	JMP LP	
		0910 :		
3AE7-	AE 60 FA	0920	LDX TEMP	: BRK SET TO THIS ADDRESS
3AEA-	E0 40	0930	CPX #40	: AT THE END OF PRCHA THERE
3AEC-	B0 04	0940	BCS RTN	: IS A TEST IF BRK IS PUSHED
3AEE-	86 FA	0950	STX #POINTL	: IF SO. THEN THIS SUBROUTINE
3AF0-	A2 41	0960	LDX #41	: IS THE END OF A JSR PRCHA
3AF2-	60	0970	RTS	
		0980 :		
3AF3-	A9 0B	0990	LDA #0B	: PERFORM THE SHOT
3AF5-	20 A3 FE	1000	JSR PRCHA	
3AF8-	A9 2A	1010	LDA #2A	: '*' IS THE BULLIT
3AFA-	20 A3 FE	1020	JSR PRCHA	
3AFD-	A9 08	1030	LDA #8	: BULLIT GOES UP
3AFF-	20 A3 FE	1040	JSR PRCHA	: TILL IT REACHES
3B02-	A9 20	1050	LDA #20	: THE WALL
3B04-	20 A3 FE	1060	JSR PRCHA	
3B07-	A9 08	1070	LDA #8	
3B09-	4C A3 FE	1080	JMP PRCHA	
		1090 :		
3B0C-	A9 08	1100	LDA #8	: BACK TO LAST PRINTED 'U'
3B0E-	20 A3 FE	1110	JSR PRCHA	
3B11-	A9 55	1120	LDA #55	: THIS HERE MOVES
3B13-	20 A3 FE	1130	JSR PRCHA	: BULLIT UP
3B16-	A9 08	1140	LDA #8	: ONE LINE
3B18-	20 A3 FE	1150	JSR PRCHA	
3B1B-	A0 0A	1160	LDY #0A	
3B1D-	20 F3 3A	1170	JSR OMHOOG	
3B20-	88	1180	DEY	
3B21-	D0 FA	1190	BNE FIRE	
3B23-	A9 01	1200	LDA #1	: BOTTOM LINE
3B25-	85 02	1210	STA *WINST	: WORTH ONE POINT
3B27-	85 03	1220	STA *DWNEXT	: ALSO EXTRA <LF>
3B29-	A9 44	1230	LDA #44	: BOTTOM LINE IS ON
3B2B-	85 FB	1240	STA *POINTH	: PAGE 44 IN MEMORY
3B2D-	20 F3 3A	1250	JSR OMHOOG	: MOVE INTO LINE OF WALL
3B30-	A0 00	1260	LDY #0	
3B32-	B1 FA	1270	LDA (POINTL).Y	
3B34-	C9 3D	1280	CMP #3D	: IS THERE A STONE ('=')
3B36-	F0 2E	1290	BEQ STAND	: YES. PRINT TOTAL
3B38-	E6 02	1300	INC *WINST	
3B3A-	E6 02	1310	INC *WINST	: NO. MORE POINTS TO WIN
3B3C-	A5 02	1320	LDA *WINST	
3B3E-	C9 09	1330	CMP #9	: ONLY 1.3.5.7.10 POINTS
3B40-	D0 05	1340	BNE OK	: TO WIN
3B42-	18	1350	CLC	: I DIDN'T WANT TO USE
3B43-	69 07	1360	ADC #7	: DECIMAL CALCULATION
3B45-	85 02	1370	STA *WINST	
3B47-	C6 FB	1380	DEC *POINTH	: LOOK ON NEXT LINE
3B49-	A5 FB	1390	LDA *POINTH	: FOR STONE ('=')
3B4B-	C9 3F	1400	CMP #3F	: I.E. CHECK MEMORY
3B4D-	F0 05	1410	BEQ STRPNT	: NO. MINUS POINT
3B4F-	E6 03	1420	INC *DWNEXT	: NEXT LINE MEANS EXTRA <LF>
3B51-	4C 2D 3B	1430	JMP GET	
3B54-	A9 00	1440	LDA #0	: THIS MEANS A HOLE

3B56-	85	02	1450		STA *WINST	: IN THE WALL SO 1
3B58-	38		1460		SEC	: MINUS POINT
3B59-	F8		1470		SED	
3B5A-	A5	00	1480		LDA *PNT1	
3B5C-	E9	01	1490		SBC #1	
3B5E-	85	00	1500		STA *PNT1	
3B60-	A5	01	1510		LDA *PNT2	
3B62-	E9	00	1520		SBC #0	
3B64-	85	01	1530		STA *PNT2	
3B66-	A9	20	1540	STAND	LDA #\$20	: THIS HERE PRINTS
3B68-	91	FA	1550		STA (POINTL).Y	
3B6A-	F8		1560		SED	: THE TOTAL POINTS
3B6B-	18		1570		CLC	: YOU HAVE WON
3B6C-	A5	00	1580		LDA *PNT1	
3B6E-	65	02	1590		ADC *WINST	
3B70-	85	00	1600		STA *PNT1	
3B72-	A5	01	1610		LDA *PNT2	
3B74-	69	00	1620		ADC #0	
3B76-	85	01	1630		STA *PNT2	
3B78-	D8		1640		CLD	
3B79-	A0	0A	1650		LDY #\$0A	: GO DOWN TO POSITION
3B7B-	20	6F	1660	DWN	JSR LFEED	: WHERE CANON WAS LEFT
3B7E-	88		1670		DEY	
3B7F-	D0	FA	1680		BNE DWN	
3B81-	20	BF	1690		JSR PRSTND	: PRINT 'STAND: '
3B84-	A5	01	1700		LDA *PNT2	
3B86-	20	92	1710		JSR PRBYT	
3B89-	AD	00	1720		LDA PNT1	
3B8C-	20	92	1730		JSR PRBYT	
3B8F-	A0	08	1740		LDY #8	
3B91-	A9	FF	1750	DELAY	LDA #\$FF	: WAIT FEW SECONDS
3B93-	8D	F7	1760		STA CNTD	: TO READ TOTAL
3B96-	2C	D5	1770	DEL	BIT RDFLAG	
3B99-	10	FB	1780		BPL DEL	
3B9B-	88		1790		DEY	
3B9C-	D0	F3	1800		BNE DELAY	
3B9E-	A9	1A	1810	BLNK	LDA #\$1A	: REMOVE 'STAND:'
3BA0-	20	A3	1820		JSR PRCHA	
3BA3-	A9	0D	1830		LDA #\$0D	
3BA5-	20	A3	1840		JSR PRCHA	
3BA8-	A4	03	1850		LDY *DWNEXT	: EXTRA (LF) TO GO
3BAA-	20	6F	1860	ZAK	JSR LFEED	: TO LINE 16
3BAD-	88		1870		DEY	
3BAE-	D0	FA	1880		BNE ZAK	
3BB0-	A4	FA	1890		LDY *POINTL	
3BB2-	A9	20	1900	POSIS	LDA #\$20	: RETURN TO POSITION
3BB4-	20	A3	1910		JSR PRCHA	
3BB7-	88		1920		DEY	: OF THE CANON
3BB8-	D0	F8	1930		BNE POSIS	
3BBA-	A6	FA	1940		LDX *POINTL	
3BBC-	4C	DD	1950		JMP CONT	
			1960	:		
3BBF-	A9	0D	1970	PRSTND	LDA #\$0D	: THIS SHOULD EXPLAINE
3BC1-	20	A3	1980		JSR PRCHA	: IT SELF
3BC4-	A9	53	1990		LDA #\$53	
3BC6-	20	A3	2000		JSR PRCHA	
3BC9-	A9	54	2010		LDA #\$54	
3BCB-	20	A3	2020		JSR PRCHA	
3BCE-	A9	41	2030		LDA #\$41	
3BD0-	20	A3	2040		JSR PRCHA	
3BD3-	A9	4E	2050		LDA #\$4E	
3BD5-	20	A3	2060		JSR PRCHA	
3BD8-	A9	44	2070		LDA #\$44	
3BDA-	20	A3	2080		JSR PRCHA	
3BDD-	A9	3A	2090		LDA #\$3A	
3BDF-	20	A3	2100		JSR PRCHA	
3BE2-	A9	20	2110		LDA #\$20	
3BE4-	4C	A3	2120		JMP PRCHA	
			2130	:		
			2140		.EN	


```

10 REM MAANLANDER
20 REM AFKOMSTIG VAN M. VAN HINTUM
30 PRINT"U ZIT IN UW MAANLANDER OP EEN HOOGTE"
40 PRINT"VAN 2500 METER VAN DE MAAN. UW SNELHEID"
50 PRINT"BEDRAAGT 500 M/SEC, EN U HEEFT NOG 800 LITER"
60 PRINT"BRANDSTOF. DOOR GEBRUIK TE MAKEN VAN BRANDSTOF,"
70 PRINT"DIENT U UW SNELHEID ZODANIG TE VERLAGEN,"
80 PRINT"DAT U MET EEN VEILIGE SNELHEID OP HET"
90 PRINT"MAANOPPERVLAK TERECHTKOMT. SUCCES!!!!!!!!!!"
100 LETS=500
110 LETH=2500
120 LETK=600
130 PRINT
140 PRINT"HOEVEEL LITER BRANDSTOF WILT U GEBRUIKEN ";
150 INPUT I
160 IF I >= 0 THEN 220
170 PRINT
180 PRINT"HO, HO, U KUNT GEEN BRANDSTOF AAN DE"
190 PRINT"RUIMTE ONTTREKKEN ! WE VERONDERSTELLEN"
200 PRINT"DUS DAT U GEEN BRANDSTOF WILT GEBRUIKEN."
210 LET I = 0
220 IF I <= 75 THEN 280
230 PRINT
240 PRINT"LAST VAN HEIMWEE, HM ? TERUG NAAR MOEDER"
250 PRINT"AARDE ? NOU, DAT GAAT NIET DOOR, WE GEVEN"
260 PRINT"U MAXIMAAL 75 LITER PER KEER !"
270 LET I = 75
280 IF I < K THEN 330
290 PRINT
300 PRINT"ZOVEEL BRANDSTOF HEEFT U NIET MEER !"
310 PRINT"WE ZULLEN U ALLES GEVEN, WAT U HEEFT !"
320 LET I = K
330 LET K = K - 1
340 LETH = H - S + (I - 5) / 2
350 LETS = S - I + 5
360 IF INT(H + 0.5) <> 0 THEN 380
370 IF S <= 5 THEN 460
380 IF INT(H + 0.5) <> 0 THEN 510
390 PRINT
400 PRINT"KWAF!BOEM!CRASH!KLEUN!!!!!!!!!!"
410 PRINT"HELAAS, UW STOFFELIJKE RESTEN LIGGEN,"
420 PRINT"TESAMEN MET DE OVERBLIJFSELEN VAN UW MAANLANDER"
430 PRINT"VERSPREID OVER HET MAANOPPERVLAK."
440 PRINT"U KWAM MET EEN SNELHEID VAN ";INT(S+0.5);" M/SEC NEER."
450 GOTO 850
460 PRINT

```

```

70 PRINT"MIJN GELUKWENSEN, EEN GESLAAGDE LANDING !"
80 PRINT"U KUNT HET ANGSTZWEET NU VAN UW VOORHOOFD VEGEN"
90 PRINT"EN EEN WANDELINGETJE MAKEN OP HET MAANOPPERVLAK."
00 GOTO 890
10 IFK<>0THEN660
20 PRINT
30 PRINT"HELAAS, U HEEFT AL UW BRANDSTOF GEBRUIKT."
40 PRINT"UW BOORDCOMPUTER VOORSPELT, DAT U MET EEN"
50 PRINT"SNELHEID VAN ";INT(SQR(S*S+10*H)+0.5);
60 PRINT"M/SEC TE PLETTER ZULT VALLEN !"
70 PRINT"HET WAS LEUK U GEKEND"
80 PRINT"TE HEBBEN. MAAR U MOET MAAR ZO DENKEN: AAN ALLES"
90 PRINT"KOMT EEN EIND. VAARWEL, RUST IN VREDE!....."
00 FORJ=1TO1000
10 LETA=2.34**0.5
20 NEXTJ
30 PRINT
40 PRINT"KWAF!BOEM!CRASH!KLEUN!.....(STILTE)"
50 GOTO 850
60 IFINT(S+0.5)>0THEN810
70 IFINT(S+0.5)=0THEN750
80 PRINT
90 PRINT"BANG GEWORDEN, HM ? U HEEFT ZO STERK AFGEREMD,"
00 PRINT"DAT U ZICH OP HET OGENBLIK MET EEN SNELHEID"
10 PRINT"VAN ";INT(S+0.5);" M/SEC VAN HET MAANOPPERVLAK"
20 PRINT"VERWIJDERT. U BEVINDT ZICH NU OP ";INT(H+0.5);
30 PRINT"METER HOOGTE."
40 GOTO130
50 PRINT
60 PRINT"U DENKT ZEKER: 'VOORZICHTIGHEID IS DE MOEDER VAN "
70 PRINT"DE PORSELEINKAST', HM ?"
80 PRINT"U HANGT NAMELIJK STIL OP ";INT(H+0.5);
90 PRINT"METER HOOGTE !"
00 GOTO130
10 PRINT
20 PRINT"UW SNELHEID IS NU ";INT(S+0.5);" M/SEC."
30 PRINT"UW HOOGTE BEDRAAGT ";INT(H+0.5);" METER."
40 GOTO 130
50 PRINT
60 PRINT"DIT WAS PRUTSWERK NATUURLIJK, VOOR IEMAND"
70 PRINT"VAN UW INTELLIGENTIE. MAAR GOED, WE ZULLEN"
80 PRINT"U NOG EEN KANS GEVEN."
90 PRINT
00 PRINT"WILT U HET OPNIEUW PROBEREN (JA/NEE) ";
10 INPUTQ$
20 IFQ$="JA"THEN30
30 STOP
40 END

```

JUNIOR 8 K OF JUNIOR 64 K *****

door : Hans Mooi
Scharstraat 37
1317 PL Almere

1.0

De JUNIOR-computer is in het begin als 8K systeem opgebouwd met 2 platen welke ik samen in een kastje heb ondergebracht. Hiervoor heb ik verschillende software, zoals schaakprogramma (DE 6502 KENNER nr.20), One Armed Bandit (DE 6502 KENNER nr.14) en andere kleine programma's.

Later zijn er uitbreidingen gekomen welke allemaal op eurokaart formaat zijn.

Ik heb mijn JUNIOR dusdanig aangepast dat, als ik hem op het 19 inch-rek plaats, ik een 64 K systeem heb. Haal ik de stekker van de uitbreidingskaart, dan ontstaat weer mijn oude JUNIOR.

Tevens heb ik de I/O-adressering wat economischer gededuceerd en daardoor dan een I/O-map (zoals ik dit noem) van F800 t/m FBFF.

intern: De PIA 6532 op de hoofdorint	=	FA00 - FA7F	RAM
		FA80 - FB00	I/O / Timer
extern: De VIA 6522 op de interfacekaart	=	F9F0 - F9FF	I/O / Timer
De Floppy Disk kaart	=	F800 - FB0F	
De Video kaart	=	F900 - F90F	

1.1

Oobouw

Op pen 2C heb ik standaard bus signaal SEL aangezet, dat F800 - F9FF de-codeert. Van de 512 plaatsen extern zijn er nog vele onbezet, en voor eigen toepassingen bruikbaar. Het memory heb ik zoveel mogelijk met RAM bezet zodat er snel van systeem programma (Basic, Forth, e.d.) gewisseld kan worden. De meeste hoofdprogramma's heb ik op cassette op de adressen \$2000 - \$3FFF. Ik ben van plan hier een EPROM te gaan gebruiken met bankswitching, onder te verdelen in 8K statische RAM + 8K EPROM, 0000 - 1FFF, resp. 2000 - 3FFF. Het video-geheugen van 2K RAM is bij mij 4K geworden door 2 x 6116 op elkaar gesoldeerd op de kaart te plaatsen. Pen 18 heb ik dan voor verdere decodering gebruikt.

2.0

Hardware aanpassingen: Upgrade

Onderdelen:

1 x 74LS12 1 x 74LS30 3 x 4K7 weerstanden

2.1) Doorverbinden 2a intern 2a extern van de 41612 DIN connector op de interfacekaart.

Noot: interne connector is de connector die naar de hoofdorint gaat
externe connector is de connector die naar de buskaart gaat

2.2) Veranderingen hoofdorint:

IC 2 vervalt en er komt een 2716 of 2732 voor in de plaats.

2716 : 1K voor het gecorrigeerde MON-programma

1K voor DOS Paperware 1 (Elektuur)

2732 : 1K als 2716 MON (gecorrigeerd)

1K als 2716 DOS

1K MON voor evt. 6809E

1K DOS voor evt. 6809E

Verwijder C 12 en C13. Kras 18a en 17c los: dit zit aan de bovenkant

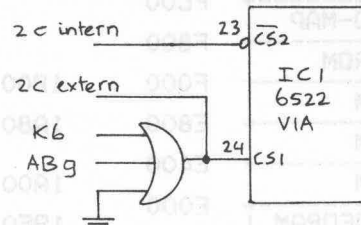
waar de toetsen zich bevinden.

Verbindt de -kant van C 13 (welke er uit is) aan +5V van IC 8 pin 14 (voor gebruik van 2732 een pull-up weerstand van 4K7).

Verbindt de +kant van C 12 (welke er uit is) met een pull-up weerstand van 4K7 aan +5V van IC 8 pin 14.

FIG.2

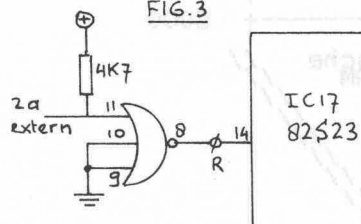
2.3) Veranderingen interfacekaart.



Haal K6 van pin 23 IC 1 los door vlakbij IC 13 pin 2 door te krassen. Leg een verbinding van IC 1 pin 23 naar 2c interne connector 41612 DIN

FIG.3

Aan de tekentafel:
Fridus Jonkman



Haal IC 13 er uit en kras verbinding van pin 6 naar pin 11 door. Kras vlakbij IC 13 pin 8 de verbinding door en leg een draadverbinding van pin 8 IC 13 naar R.

IC 13 pin 11 met een draad naar externe connector 41612 pin 2a. Leg tevens een pull-up weerstand van 2K2 van pin 2a naar +5V. 1a + 1c connector 41612 DIN extern.

2.4) Maak nu een printie met de 2 IC's 74LS30 en 74LS12 en verbindt deze op de hoofdprint.

FIG.4

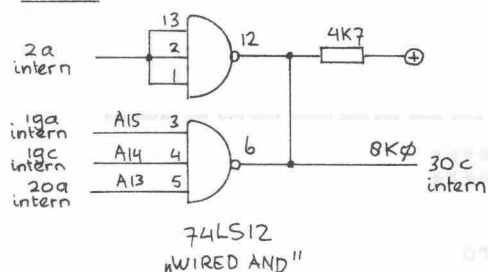
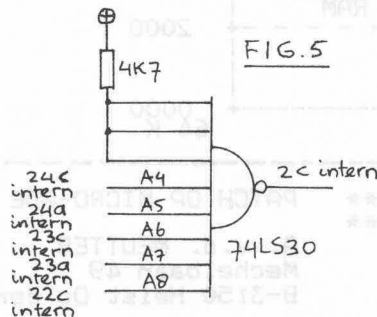


FIG.5



Uitselecteren van VIA F9F0 t/m F9FF als standaard 8K systeem.

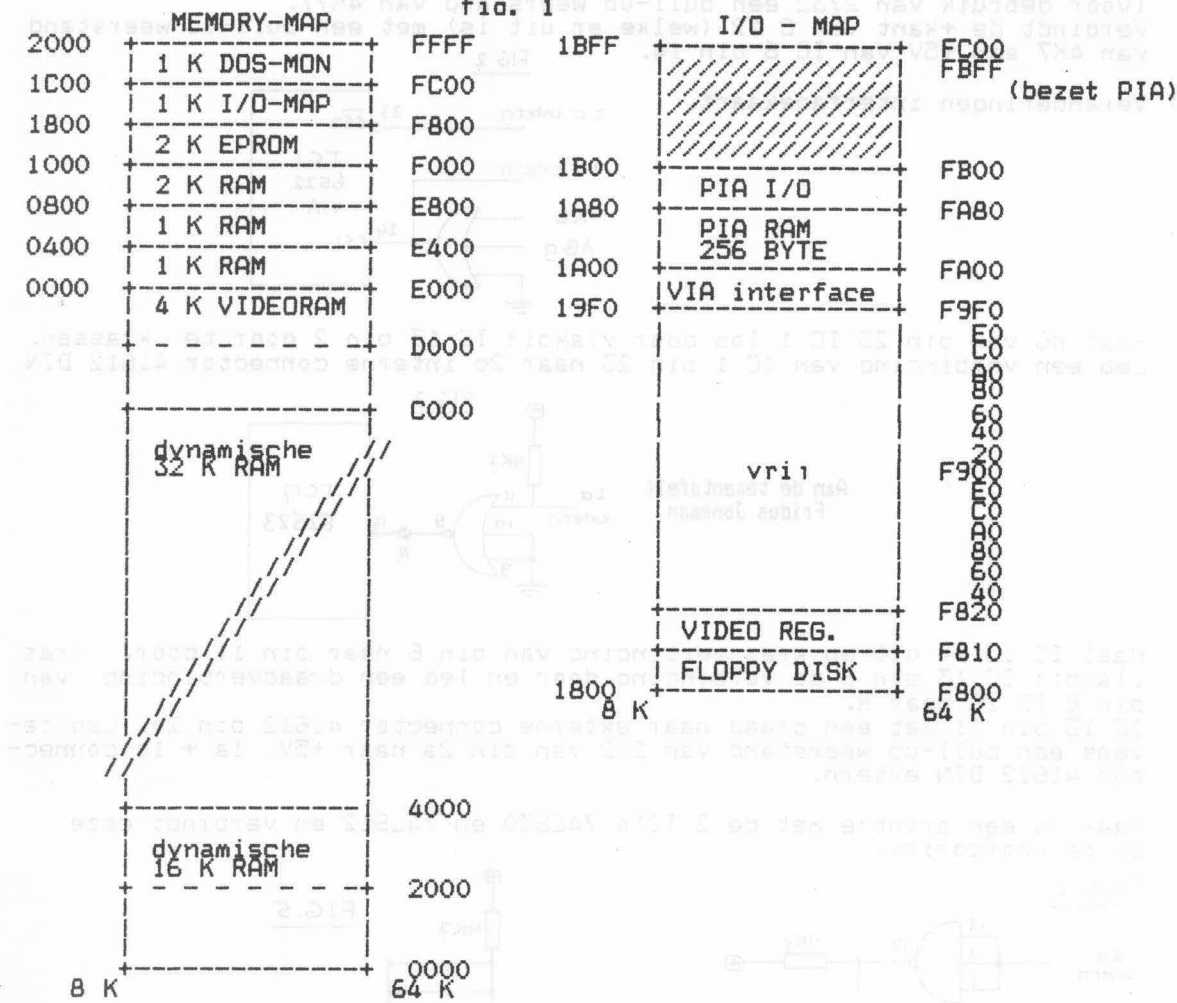
Verbindt nu op alle uitbreidingskaarten pin 2a van 41612 DIN connector met aarde. pin 4a.c.

Bij het verwijderen van alle uitbreidingskaarten zal de JUNIOR als 8K systeem werken (zoals vroeger).

De eerste uitbreidingskaart zal behouden op plaats 0000 t/m 0200 moeten hebben. anders is er op deze plaats niet de nodige RAM aanwezig.

1FFF/ FFFF	+	-----	+	8 K STANDAARD SYSTEEM
	!	1 K MON	!	
	+	I/O PIA+VIA	+	
	!	2 K EPROM TM	!	
	+	2 K RAM 6116	+	
	!	1 K RAM	!	
	+	1 K RAM	+	
	!		!	
0000/ D000	+	-----	+	MEMORY MAP (64 K)/8 K

fig. 1



```
***** PATCH OP MICRO-ADE *****
*****                               *****
A. v.d. MEUTTER
Mechelbaan 49
B-3150 Heist Op den Berg
```

Do Micro-ADE heb ik een kleine patch gemaakt, en wel voor LIST. Normaal kan je LT gebruiken voor listing zonder regelnummer. Je kan alleen terug regelnummers krijgen door PASS 2 uit te voeren. Dit is niet langer zo !

Nu werk je als volgt :

LI of LIST	geeft regelnummers
LT	geeft zonder nummers
L	blijft zoals laatste keuze.

Je kan dus op ieder moment kiezen tussen LI en LT.

```

30EE EA
30EF EA
30F0 AD 01 01 LDA $0101
30F3 C9 54 CMPIM 'T
30F5 F0 0A BEQ CHANGE
30F7 C9 49 CMPIM 'I dit is nieuw
30F9 F0 06 BEQ CHANGE
30FB EA NOP
30FC EA NOP
30FD 20 67 23 JSR LIST execute LIST
3100 60 RTS
3101 85 4D STAZ $4D
3103 F0 F8 BEQ $31FD

```

De enige wijziging is van 30F0 tot 30FC.

=====

C. B. M. 64 KERNAL TEST ROUTINE

=====

G. v. Roekel
v.d. Palmstraat 11C
3135 LK VLAARDINGEN

Met een hulproutine is het mogelijk om de BASIC en MONITOR ROM te kopiëren naar RAM. Hieraan is wel een maar verbonden. Men kan wel alleen de BASIC ROM naar RAM kopiëren, maar niet alleen de MONITOR van ROM naar RAM. Voor het kopiëren van de MONITOR ROM dient u eerst de BASIC ROM naar RAM te kopiëren. Met onderstaand programma wordt de KERNAL routine gezet, waarbij u de keuze heeft om alleen de BASIC ROM te kopiëren of BASIC en MONITOR ROM.

KERNAL ROUTINE

```
100 PRINT"1 A000-BFFF IN RAM"
110 PRINT"2 A000-BFFF + E000-FFFF IN RAM"
120 PRINT"MAAK U KEUZE":INPUT A
130 PRINT"EVEN GEDULD A.U.B."
140 IFA=1THENGOSUB170
150 IFA=2THENGOSUB170:GOSUB190
160 END
170 FORJ=40960 TO 49151:POKEJ,PEEK(J)
180 NEXTJ:POKE1,54:RETURN
190 FORJ=57344 TO 65535:POKEJ,PEEK(J)
200 NEXTJ:POKE1,53:RETURN
```

TEST KERNAL

```
100 FORJ=41848 TO 41852
110 READY:POKEJ,Y:NEXT
120 DATA 75,76,65,65,82
130 FORI=61657 TO 61674
140 READX:POKEI,X:NEXT
150 DATA 68,82,85,75,32,80,76,65,89
160 DATA 32,84,79,69,84,83,32,73,206
```

Indien u bij het eerste programma heeft gekozen voor nummer 2 dan zal na het 'RUNNEN' van het tweede programma het woord 'READY' vervangen zijn door het woord 'KLAAR'. En als u het woord 'LOAD' intoetst zal op het scherm de tekst 'DRUK PLAY TOETS IN' i.p.v. 'PRESS PLAY ON TAPE'. Mocht blijken dat het niet werkt, controleert u dan of beide programma's goed zijn ingetoetst. Anders is het noodzakelijk om met uw COMMODORE 64 terug te gaan naar de winkelier. Het is namelijk gebleken dat er met een aantal machines geen KERNAL routine mogelijk was. Dit uitte zich over het algemeen in het niet goed functioneren van een aantal machinetaal programma's welke gebruikt maakte van deze KERNAL routine.



Versneld laden van BASIC-programma's op de SENIOR

Door: Gert van Dobroek
Hooglanden 20
9801 LB Zuidhorn
Tel.: 05940 - 5627.

Bij het wegschrijven van BASIC-programma's naar schijf d.m.v. het commando "SAVE" wordt er een ASCII file aangemaakt. Bij het inlezen van een dergelijke file d.m.v. het commando "LOAD" vindt er tijdens het inlezen een omzetting plaats. Het gevolg hiervan is dat het laden van grote programma's (16K) vele minuten vergt. Dit is te ondervangen door buiten BASIC om een memory dump van het programma te maken. Dit kan als volgt gedaan worden:

1. Vraag (binnen BASIC) het eindadres +1 van het programma op d.m.v. de opdrachten:
"CLEAR"
"PRINT HEX\$ (&nnnn-FRE(0))"
waarbij nnnn het hexadecimale adres van de hoogste RAM-locatie is. Indien BASIC dit adres bij de koude start zelf bepaald heeft, is dit maximaal HEX BFFF.
Het beginadres van het programma is HEX 4D00.
2. Ga uit BASIC met de opdracht "MON". Schrijf nu het onder punt 1. gevonden geheugengebied weg met de monitoropdracht "D". Schrijf bovendien pagina 0 (HEX 0000 t/m HEX 00FF) in dezelfde file weg.

Het inlezen van een op deze manier weggeschreven BASIC-programma gaat nu als volgt:

- A. Laad BASIC, doch start deze niet (of verlaat BASIC met de opdracht "MON").
- B. Laad het BASIC-programma met behulp van het monitorcommando "L".
- C. Start BASIC met een warme start d.m.v. het monitorcommando "6".

ELEKTUURS SAMSOM-65 DOS COMPUTER

In het voorjaar 1985 brengt Elektuur haar SAMSOM-65 DOS-computer uit. Op maandagavond 3 sept. 1984 werd ons deze nieuwe zelfbouw computer oedemonstreerd. De SAMSOM-65 is opgebouwd uit reeds bestaande eurokaarten, zoals de CPU- en VDU-kaart, dynamische Ramkaart en/of universele geheugenkaart, en de voor het OS65D Disk Operating System V3.3 bedoelde controllerkaart. Vergeleken met de JUNIOR, waarvoor erg veel sets boeken werden verkocht, zijn er twee in het oog lopende en aantrekkelijke verschillen. Ten eerste is de computer opgebouwd uit eurokaarten, zodat het geheel gemakkelijk kan worden ondergebracht in een 19" rek. Ten tweede is er gezorgd voor gebruikersvriendelijke software voor het monitorgebeuren. Hierin staan full screen editing en wordprocessing centraal, naast een aantal aantal utilities. De "Kenners" zullen in deze software, die tegen geringe vergoeding door Elektuur op uw diskette wordt geschreven, delen van de populaire Micro-ADE herkennen. Voor de SAMSOM-65 heeft Elektuur al specifieke toepassingen in ontwikkeling. De redactie zal de komst van deze computer alvast ondersteunen met een publikatie die het OS65D in belangrijke mate gebruikersvriendelijker zal maken. We zullen de toekomstige gebruikers ervan met de ervaringen die we in huis hebben een goed onderdak kunnen verschaffen.

KORREKTIE

IN HET ARTIKEL "USING FORTH WITH THE 6502" UIT NR. 30 STAAT EEN FOUT. DIE IK HIERBIJ DANK ZIJ EEN BETER INZICHT IN FORTH (VIA DE METHODE VALLEN EN OPSTAAN VERKREGEN) KAN KORRIGEREN.
EEN DEEL VAN DE TEKST UIT HET ARTIKEL IS OPNIEUW AFGE-DRUKT:

HET TOEVOEGEN VAN WOORDEN AAN EEN BESTAANDE VOCABULARY. WAARBIJ DEZE IN HET SYSTEEM "GELOCKED" MOETEN WORDEN.

GAAT ALS VOLGT:

FORTH DEFINITIONS DECIMAL

LATEST 12 +ORIGIN !

HERE 28 +ORIGIN !

HERE 30 +ORIGIN !

HERE FENCE !

WANNEER EEN AANTAL WOORDEN, DIE EERDER ZIJN "GELOCKED" IN HET SYSTEEM, MOETEN WORDEN VERWIJDERD, DAN MOET HET VOL-DE WORDEN UITGEVOERD:

FORTH DECIMAL

' AAA NFA 12 +ORIGIN !

YYY DEFINITIONS

' BBB NFA 28 +ORIGIN !

' BBB NFA 30 +ORIGIN !

' BBB NFA FENCE !

' ZZZ 6 + 32 +ORIGIN !

FORGET BBB

WAARIN: AAA HET LAATSTE WOORD IS VAN DE FORTH-VOCABULARY
BBB HET EERSTE WOORD IS DAT UIT HET SYSTEEM VER-
DWIJNT

YYY DE VOCABULARY IS WAARIN BBB VOORKOMT

(DEZE REGEL IS ALLEEN NODIG INDIEN BBB VAN EEN ANDERE VOCABULARY DEEL UITMAAKT DAN FORTH: HET WOORD DEFINITIONS IS ALLEN NODIG BIJ GEBRUIK VAN FORTH VOLGENS DE 79-STANDAARD)

ZZZ DE LAATST GEDEFINIEERDE VOCABULARY IS DIE IN HET SYSTEEM BLIJFT

(DEZE REGEL IS ALLEEN NODIG WANNEER EEN OF MEER VOCABULARIES GEHEEL VERDWIJNEN).

WEL DIEN BEDACHT TE WORDEN DAT ALLEN EEN OF MEER WOORDEN, DIE EEN AANEENSLUITENDE REEKS VORMEN EN DIE HET LAATST AAN HET SYSTEEM ZIJN TOEGEVOEGD OP DEZE WIJZE VERWIJDERD KUNNEN WORDEN.

FRIDUS JONKMAN

@

BUG IN DOS UITBREIDING VAN ELEKTUUR, SEPTEMBER 1984.

Gert Klein, Diedenwee 119, 6706 CM Wageningen meldt ons: Bij het OHIO DOS moet je eerst een file in de directory creëren alvorens je de betreffende file kunt wegschrijven. De door Elektuur gepubliceerde patch probeert dit te omzeilen. In een bepaald en helaas veel voorkomend geval leidt dit tot tamelijk ruineuze resultaten. Wat is nu het geval: wanneer je nu een file wilt wegschrijven en er is nog geen entry in de directory aanwezig dan creëert de genoemde patch zie zelf. De ruimte is echter precies groot genoeg om de file op floo te schrijven. Wanneer je later de file gaat uitbreiden, dan maakt de patch de benodigde ruimte niet groter. Helaas is de OHIO DOS niet slim genoeg om dat zelf te ontdekken en probeert dus de file weg te schrijven in een te kleine ruimte. Op zich is dat nog niet zo erg, maar het resultaat is wel dat de file op de floo grondig in de vernieling wordt geholpen.

STUUR DE REDAKTIE UW ZELF ONTWIKKELDE SOFTWARE/HARDWARE !



all-round

All-round Systemen Nederland B.V., Stationsweg 23, 2182 BA Hillegom
postbus 212, 2180 AE Hillegom. tel. 02520-19544. telex 41443 allro nl.

WIJ LEVEREN SNEL EN TEGEN ZEER GUNSTIGE PRIJZEN:

Diskettes

van verschillende merken zowel 8" als 5 $\frac{1}{4}$ ". Tevens formatted en cleaning diskettes.

Papier

voor kleingebruikers leveren wij dozen van 1000 vel 38 cm.

Inktlinten

linten en kassettes voor alle gangbare printers.

Computermeubilair

o.a. ideale tafeltjes om een compleet micro systeem op te plaatsen.

Data ~safe's

een mini safe voor 80 x 5 $\frac{1}{4}$ ", of 30 x 8" diskettes.

Opbergssystemen

o.a. voor diskettes, tapes, kassettes en alle maten output.

Geluiddempende kappen

brengen het geluid van uw printer 75 tot 90% terug.

Alles voor uw computer



all-round

Micro-ADE Assembler/Disassembler/Texteditor
for 6502 (8K)

SYSTEM DESCRIPTION

The Micro-ADE system is designed for use with any 6502 microcomputer and consist of three major programs as well as a number of utility programs. The major programs are an assembler, a disassembler, and a text editor.

The assembler is used to create machine executable code for the 6502 from a symbolic input source program. Small programs can be created and tested directly in memory. Larger programs may be written using cassette tapes for source input and object output.

The disassembler is used to list executable 6502 machine code in the symbolic assembler source format. Symbols are generated if they are defined in the symbol table.

The text editor is used to create source programs in the format required for the assembler. It contains the necessary routines for easy manipulation of text data in memory or from cassette files.

The minimum system configuration for full use of all Micro-ADE features consists of a 6502 CPU, 8K of random access memory, 2 cassette recorders with start/stop control, and an ASCII input/output device. It is possible to use all parts of the system in a restricted way with less memory and a single manually operated cassette recorder.

For some months past our club has obtained copyrights on Micro-ADE of Peter Jennings from Micro-Ware Ltd. We are allowed to place Micro-ADE at the disposal of the members of our club.

The original 4K Micro-Ade has been patched up to 8K by Sebo Woldringh, with commands like Append Source Files, Insert Lines, Set/Reset Form Feed Flag, Chosse Lines, List, Move Lines, Load ASCII-format Files, Set/Reset Page Mode, List Used Memory, Duplicate Files, Save Source Files, Print Symbol/XREF-tables, Set/Reset Page-per-File/EJECT Flag, Search Line, Print/Change String, Assemble/Execute, Set Lines/Screen and Lines/Page, Disassemble, List without line numbers, Save chosen Lines, Save ASCII-format Files, etc.

From these patches W.L. van Pelt composed an 8K source-listing, complete with comments.

Manual+Command Review: Fl. 35,00. Source Listing 8K-version: Fl. 65,00. Original 4K Source-Listing: Fl. 30,00. Cassettes in KIM/JUNIOR hypertape: 4K KIM = Fl. 12,50, 8K KIM = FL. 12,50, 8K JUNIOR = Fl.12,50.

To be paid with Eurocheque, or on the postal account 841433 from W.L. van Pelt, Krimpen a.d. IJssel, The Netherlands. No other payments allowed.